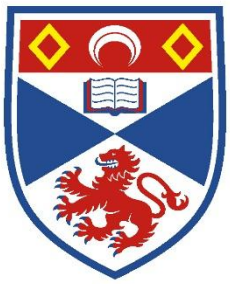
Linked Data – Exposing Uncertainty

Tom S. Dalton



University of
St Andrews

This dissertation is submitted in
partial fulfilment for the degree of BSc (Hons)
at the
University of St Andrews

10 April 2015

Supervisor: Dr Graham Kirby

“I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this dissertation is 19,485 words long, including project specification and plan.

In submitting this dissertation to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work."

Table of Contents

1 Abstract

This research focuses on approaches to structure and query linked data sets created by an idealised linkage process. The data linkage process is able to suggest multiple possible linkages with estimates for the certainty of each suggested linkage. This paper defines a data structure that is able to store the linked data set and the associated uncertainty, in addition to the provenance of each linkage. Approaches to querying the data structure and providing textual justifications for query results, are also defined for a number of genealogical relationships. This work also considers the value of data pertaining to social constructions (e.g. marriage, adoption) and its use in supporting and interfering genealogical relationships. The conclusion explores the implications of this research for the future development of the defined idealised linkage approach.

2 Statement of Project Aim

The aim of this project is to explore new ways that linked genealogical data sets with uncertainty can be expressed, stored and represented. These linked data sets arise from taking existing discrete data sets and combining them together to create new linked data sets. These can be very useful when performing research that spans multiple domains, but where we believe that inferences can be made about one domain by knowing things about the other. However, in the process of creating linked data sets, uncertainty often arises. When we decide a representation of an entity in each data set represents the same real world entity, we are able to link or associate the data from both data sets with the single entity. When this happens we can rarely be certain that the linkage we are creating is correct; we may find there are multiple possible ways in which the two data sets can be linked, resulting in uncertainty of the linkage we have made. In this work we assume that we have an idealised linkage algorithm that is able to detail the uncertainty and multiple linkage possibilities; therefore focusing on how to represent this to the user.

We are also interested in the provenance of the linkage possibilities being found. Any linkage that is suggested is based upon an underlying set of source records from within the data sets being linked. Therefore we assume that the idealised linkage process will output these alongside the linkage possibilities; these will feature in the way we represent linkages to the user.

This will involve the defining of a data structure that is sufficiently expressive to represent all logical genealogical relationships based upon a range of record sources. Despite being reliant on source records the structure itself should be agnostic of any particular record format or topology. Instead it should be based upon real logical reference points, for example in the genealogical case that a person can have only two biological parents or that a person can potentially produce offspring with any number of distinct partners.

Once a sufficiently expressive structure has been defined, then ways of restricting the structure to allow a number of pedigrees[1] and likely relationships to be surmised from the data structure will be devised. This will be necessary to variably limit the complexity of the structure while maintaining sufficient expressiveness. Beyond this, ways of querying the structure will also be created. These will need to be able to give consideration to the uncertainties that exist in the structure, inherent from the linkage process, as well as identify which linkages are able to co-exist in a realistic pedigree.

The project will also explore ways to express textually the justifications and provenance behind the set of results for each query. In evaluation a consideration will be made of the approaches taken, their suitability, efficiency and the wider implications for other linkage related domains and the required approach of the underlying idealised linkage process.

[1] The recorded ancestry or lineage of a person or family.

3 Background

Before detailing the particulars of the research undertaken, it is first important to define the wider field in which the work sits, its terminology, and the implications of the field. This will provide context for the findings being presented throughout the remainder of this work.

3.1 The Origins of the Data to be Represented

The data underlying the models in this work is from the domain of genealogy. Genealogy involves the study of populations over time, the changes in individuals, and their relationships and interactions that can be recorded and considered. The data can come from a range of sources, for example birth, death, and marriage certificates; christening, census, and health care records, to name a few. It is possible to consider that one could construct a genealogical structure, a family tree per se, using the information found on the different source documents.

One could take a birth record and identify the name of the child and of the parents. From this, a census record from a decade previous could identify one of the parents named as a child within a family schedule,[2] on this record the names of the parent's sibling could also be identified. From this, a birth record could then be found with one of the siblings named as a parent thus identifying the child named on the second birth certificate as a first cousin of the initial child. From this simple example we can see that we can traverse genealogical structures built upon a range of source records. However, making 'links' across the source records, to create traversable data structures, is not a simple task.

The basis of creating links is to identify multiple entries in the source records where the same individual appears. If every individual had a unique identifier that was present across all the source documents then this task would be very simple, but in the real world, records tend to lack this. This means that the decision of whether two entries pertain to the same individual has to be made based upon the available information that is common across the records being compared, for example, first name, surname, date of birth, birth place, and occupation. Given that names alone cannot be seen as unique identifiers in a large scale population it is required to consider each different piece of information in the source records to make 'links' between different entries. When these links are being made there is a degree of uncertainty due to the difficulty in uniquely identifying individuals. An estimate can be given for this uncertainty based upon the similarity of the compared entries. Greater certainty can be given to a linkage if further records exist that support the linkage are also found. For example, take three records detailing the information forename (F), surname (S), D.O.B. (D), hospital number (H), and birthplace (B). The first record is a hospital record detailing F, S, D and H; the second a birth certificate detailing F, S, D and B; and the third record from an admin database detailing S, H and B as laid out in figure 1.

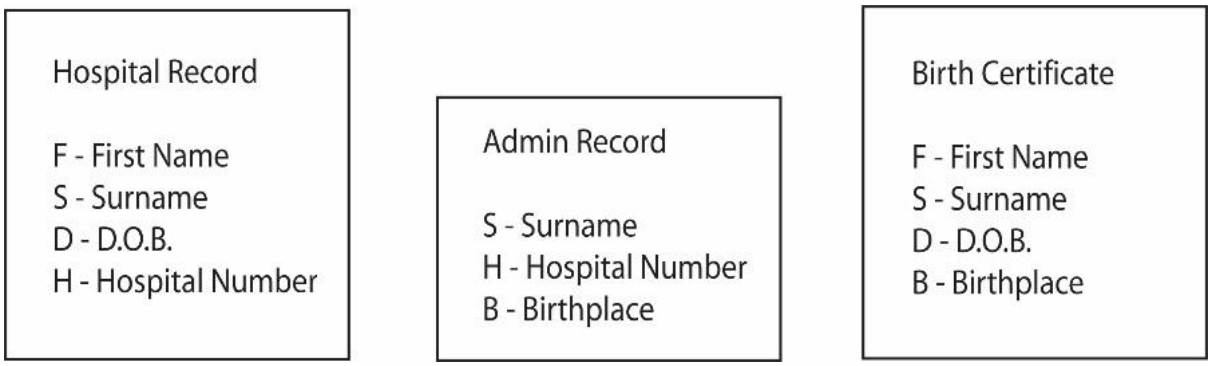


Figure 1 – Set of records for example linkage problem. The letters given are used as abbreviations within the text.

[2] A schedule is a record that specifies all the members of a household at the time of a census.

A linkage can be made between the hospital record and the birth record based on being able to match the content of F, S and D; this linkage can then be further supported if a record found in the admin database makes a connection between H and B. The admin record, in this case, can be considered a supporting document to the initial linkage which increases the certainty in the linkage that has already been made.

The idea of matching data across multiple record sources to identify links that form data structures is termed data linkage. In this field, research is occurring across a range of domains, taking the above defined process, which is simple to imagine a human performing, and enabling a computer to do the same. Such research stretches back decades (Dunn, 1946; Fellegi & Sunter, 1969; Newcombe, et al., 1986) with the value and far reaching potentials of linkage being identified even as far back as Dunn's research. He introduced an idea of a person's book of life linking together all the details about one person into a single volume and saw data linkage as the tool to enable this. Linkage approaches have evolved and their accuracy improved over the years, and large scale digitised databases, which are now becoming available, are becoming increasingly valuable. The idea of linkage laid out above defines the general idea of linkage, i.e. the matching of common entities across source records and their linking together to create data structures.

The research laid out in this dissertation however looks at representing the output of a linkage process that goes beyond a modern linkage algorithm (which simply outputs a single best fit set of links between entities) and works with an idealised linkage process output. This output identifies the source records that support each given link, gives an estimate for the certainty of the link and also can return multiple possible links when it is not possible to give a definitive linkage on a particular entity.

3.2 Wider Context of Research

Research into data linkage can be seen across a range of longitudinal projects around the globe. Research notable within the field can be seen over the past two decades in the Western Australia Data Linkage System (Holman, et al., 2008) and the Rochester epidemiology project (Melton, 1996). More recent work, of greater influence to the research laid out in this paper, is the Digitising Scotland project which is developing approaches to linking large scale data sets of birth, marriage, and death records, with the aim of building genealogical structures. The project is interested in exploring ways to calculate linkage certainty and provenance and allowing multiple possible potentially conflicting linkage solutions to co-exist. The research presented here considers what form the output of this idealised linkage process will take as to provide an intelligible and useful end representation of the data. Building an understanding of the output of the idealised linkage process will have benefits for further research centred on linkage process design.

3.3 Probabilistic Databases

Research also exists regarding probabilistic databases which look to represent database entities that cannot be deterministically classified as probabilities. The probabilistic approaches taken in these database style approaches (Aggarwal, 2009; Barbará, 1992) are rooted in the realisation of real world data and entities being uncertain - as has been already been alluded to in the case of data linkage and laid out in section 3.1. Research into uncertainty and linkage will inherently produce data where entities (as well as the linkages between them) are uncertain but the direct usage of probabilistic databases is not a possibility due to their inability to handle multiple possibilities for a value well. However, some of the approaches that probabilistic databases use in handling uncertainty may provide useful ideas which apply to parts of this work.

3.4 Application to Other Fields

The application of this research to fields beyond genealogy is also worth considering. Genealogy is often used as a good testing ground for new approaches to linkage; the value of a linkage process that considers uncertainty and its represented could have considerable implications for fields such as medical record linkage and security linkage. For example, in both cases action is likely to be taken on high health risk and flagged individuals respectively. These actions will be based upon the linkages made and therefore, in the presence of uncertainty, it is important that a quantified estimate for uncertainty is presented to the users, so that they can make better decisions, based on a better understanding of the data. The value of data which can be used with nuance has wide reaching implications for how linked data is used both locally for example, across the remit of the Administrative Data Research Centre (ADRC-S) and further afield where linked data sees commercial use.

3.5 Other Representations and Ontologies for Genealogical Data Sets

The way in which genealogical data and the relationships between entities is represented is considered by a number of different model specifications and ontologies, for example GENTECH and OWL respectively. The GENTECH model (GENTECH, 2000) focuses on how data can be stored with full expressiveness when data arises from multiple different source records and aims to facilitate better sharing of genealogical data between researchers. However, the model does not offer approaches to handling uncertainty and multiple linkage, the area where this work focuses.

The Web Ontology Language (OWL) (McGuinness & Harmelen, 2004) builds upon the Resource Description Framework (RDF) and offers a formal way to describe taxonomies and classifications in networks. There has been some work (Stevens & Stevens, 2009; Tsarkov, et al., 2008) into using the OWL model to represent genealogical relationships. Modelling population structures in this way offers some interesting benefits, for example, that edges once created may be considered in both directions. Despite this, issues are noted pertaining to too many inferences being made and a lack of ability to infer full and half relationships (Stevens & Stevens, 2009). The limitations of the OWL ontology for this project are likely to arise in the over generation of inferences, especially in light of the volume of additional links that are to be created in our proposed structures to handle uncertainty.

4 The Research

In outlining the approach taken in this research we will first discuss the development and design of the structure demonstrated by the use of a number of use cases. Once the structure has been introduced, the restrictions which can be laid on top of this will be discussed. Additionally, the approaches taken to querying the structure and the ways that meaningful information can be returned will be explored. This involves generating and structuring the returned information as well as considering the value and approach taken to providing textual justification.

4.1 Design Considerations

This section will focus on the conceptual development approach and consider a query language for the structure. The associated queries will also be discussed and defined. The scalability, intelligibility and usability of the model, for the given domain, will also be considered in evaluation using a number of varied use cases.

4.2 Approach

This work began with an initial specification that had been laid down during previous research work. The next step was to expand on this so as to allow the many possible genealogical and uncertainty permutations to be represented, while offering a structure that enables restrictions to be placed within the data structure.

4.3 The Idea

In the next sections we will explore the conceptual development of the data structure and associated interfaces. However, before doing this, a clear understanding of the high level idea that we are attempting to implement may be useful.

Consider that we have a world containing real people, that are born, live, marry, reproduce, divorce, remarry and die – in some combination and order. Ideally, we want to be able to make computer models that are a perfect representation of this real world, meaning that every entity in the real world is correctly represented in the model. These models are therefore based on data, in the form of source records, created by actions occurring in the real world. However, in a set of source records, some may be missing, the data on some of them lost, and either due to false information or typographic error, the data on some may be incorrect. Because of these issues, uncertainty arises as we pull together different data sets, meaning that doubts about the structure of our data and models begin to appear. Within our models we then need to find ways to capture the things that we do know, alongside the decisions and inferences that we make in the creation of a model. The more information, provenance, and understandings we can store about the data in a form that allows for multiple solutions to be stored in our model, the more realistic our model will be able to be later when we wish to extract information pertaining to a subset of the tree.

As we go forward we are aiming to devise data structures (i.e. models) that are able to best represent the real world. This involves holding as much information as possible in our structures pertaining to the world and being able to query them in ways that allow the possible different connections in the structure to be seen, as well as an understanding of their genealogical relevance imparted and their provenance.

4.4 Start Point – Initial Interfaces

The structures that we define are underwritten by interfaces. These describe the different objects in the system and the information they are expected to hold, including which other objects they are connected to.

The initial interfaces for the data, which outlined person and partnership objects, are shown in figure 2. These interfaces have been created through a number of research projects, most recently in research in which I was involved in the summer of 2014.

In this initial implementation, a person has a number of partnerships of which they are a member and a single partnership of which they are the child. A person being able to be a member of multiple partnerships arises out of the real world possibility that an individual may produce offspring with a number of people over the course of a lifetime. A partnership has a single male and female member and also a list of children; a single pair of genealogical parents has been enforced mainly due to simplicity but also because the frequency of adoption by same sex couples is negligible in historical records. The assumption in the use of this structure is that any children listed are known children of both the given parents. The restrictions in this structure limit the complexity of the scenarios that can be modelled using the initial interfaces.

```
public interface IPerson {

    int getId();

    String getFirstName();

    String getSurname();

    char getSex();

    Date getBirthDate();

    String getBirthPlace();

    Date getDeathDate();

    String getDeathPlace();

    String getOccupation();

    String getDeathCause();

    List<Integer> getPartnerships();

    int getParentsPartnership();

}
```

```
public interface IPartnership
    extends Comparable<IPartnership> {

    int getId();

    int getFemalePartnerId();

    int getMalePartnerId();

    int getPartnerOf(int id);

    Date getMarriageDate();

    String getMarriagePlace();

    List<Integer> getChildIds();

}
```

Figure 2 – The initial interfaces.

It could be considered that the existing interfaces be used with a new set of assumptions to allow uncertainty to be represented. This approach would involve creating a partnership object for each possible pairing that the linkage algorithm identifies and then to place the children for each possible partnership into the list of children. However, it can be easily seen that the massive number of partnership objects, due to the number of permutations of joining together two sets of parents, would be difficult to understand and to abstract meaningful pedigrees. Furthermore maintaining the current interfaces would mean that there would be nowhere in the structure to indicate the likelihood of one linkage solution over another or the provenance and reasoning behind each edge.

From this it can be seen that the interfaces need to be opened up to allow more flexibility to represent a wider range of genealogical scenarios. The approaches discussed below first focus on creating a set of interfaces which are sufficiently expressive.

In the next section we will explore the development of the data structure by considering a set of progressively complex representative scenarios. First, however, we will outline the parts that exist within this structure, the elements they consist of and the assumptions we are making regarding real genealogical structures and the data sets being linked.

4.4.1 People and Objects

All the structures we will consider consist of people. These were defined in the original interfaces and little change occurs to the interface which stipulates that a person has a single partnership from which they are born and a list of partnerships (bearing children) which they are a member of. As detailed above a person also contains information about their name, sex, birth, death and occupation. People are represented by capitalised letters from the English alphabet throughout.

A main part of all of the data structures is also the childbearing partnership objects. These find their basis in a similar interface to the IPartnership interface laid out above but have a few notable differences, including the removal of any details pertaining to marriage, limiting the number of children objects to one and permits multiple possibly people to be listed as the mother and father in the partnership. The exact details of this interface is outlined later and the way in which it has been arrived at discussed in the next section. It is useful, however, to note at this point that childbearing partnerships not only assert a set of possible mothers and a set of possible fathers but also allow us to see pairings between parents. This is useful as it allows us to be able to be more informed about the population which will in turn allow us to make further decisions, for example, if two children are half or full siblings. Childbearing partnership objects are represented by the T shaped objects seen in the diagrams depicting the possible data structures labelled with a Greek letter and have lines connecting the three end points of the T to person objects (capitalised English letters).

Bridges are another type of object that will be used in the structure. The first type of bridge is a sibling bridge. These are supported by underlying source records that specify relational connection (i.e. siblings, cousins, etc.) rather than direct genealogical relationships (i.e. parent, child) and so arise from differing source records, such as census records where we may see a household listed on a schedule and so it can be seen that two people are siblings without knowledge of a common parents. The fact that the additional information found on sibling bridges must originate from different source records means that when we find a sibling bridge with a corresponding path across the data structure (i.e. child-parent-parent-child) means that we will be able favour more strongly a particular pedigree in light of the evidence. Also it should be noted that bridges are representative of an underlying source record and are not placed into the data structure based upon inferences made from other objects in the data structure. Sibling bridges are represented in the structure by a single line annotated with the term ‘sibling’ with lines joining either end of the bridge to sets of people.

The second type of bridge is a marriage bridge. These are again supported by underlying source records that specify a marriage between two individuals, for example, a marriage certificate. A marriage bridge is also based on a similar interface to the IPartnership interface already detailed but with the removal of details pertaining to children and now permitting multiple possibilities to be listed as the husband or wife in the partnership. Marriage bridges are depicted in the data structure in the same way as sibling bridges but are annotated instead with the term ‘marriage’.

The other type of object in the structure that is especially easy to overlook are the Links. These are used in the structure to connect together objects and people. Within them is information detailing which object and person they are linking together, the evidence (i.e. source records) that supports the link, and the estimates that represent the certainty of the link which is outputted by the underlying linkage process.

4.4.2 Assumptions

Within our data structure we make certain assumptions about the nature of genealogical relationships and the underlying data sets, these include:

- A person has one biological father and one biological mother
- Our data sets may be incomplete
- A data set does not have any duplication of relationships or people within itself
- Sibling relationships result from one shared parent or two shared parents
- The output from the linkage process will offer a number of possible linkages between entities in the data sets, however some of these will be incorrect, therefore, even in the presence of multiple connections between two individuals they may actually not be related.

4.4.3 Example diagrams

Shown below are the two types of diagram that will be used throughout the next sections. The first is used to show a data structure in which we are able to represent uncertainty and multiple possible people for a given role. For example, figure 3 depicts a scenario where A is the child with one possible mother, D, and two possible fathers, B and C.

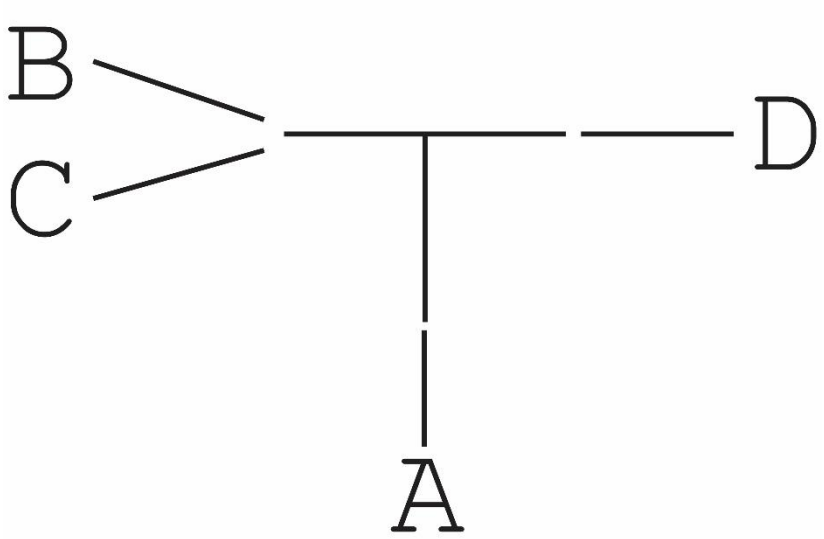


Figure 3 – An example diagram of a data structure.

The other type of diagram used is akin to a family tree. The above structure shows that there are two possible pedigrees that can be created. Figures 4 and 5 show the two family tree diagrams representing these. The family tree diagrams can be distinguished from the data structure diagrams by the thickness of the T pieces and by the lack of links/edges.

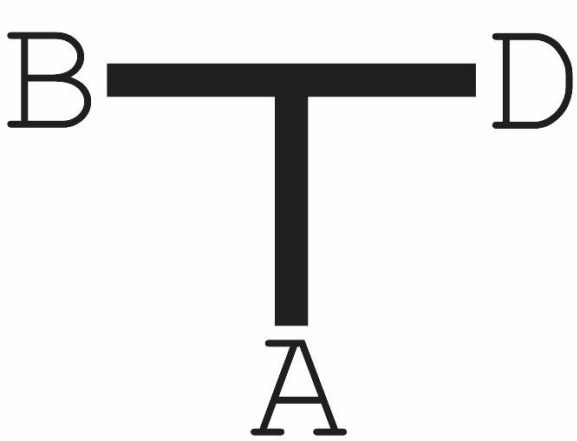


Figure 4 – An example diagram of a family tree, depicting one possible pedigree from figure 3.

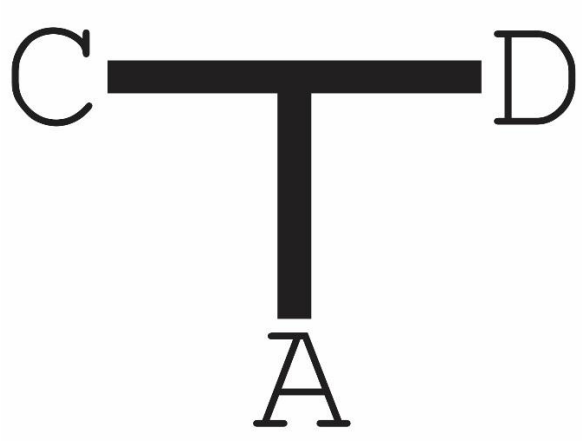


Figure 5 – An example diagram of a family tree, depicting the other possible pedigree from figure 2.

4.5 Case Studies

The following section now addresses the conceptual journey taken to create the objects and models that have been presented thus far. This is done by presenting various case studies and varying the assumptions that we make when considering the data structures.

4.5.1 Child and parents case study

Firstly we can consider a single child with a set of possible mothers and fathers. Biologically we can be certain of the child having one father and one mother, although the parents may not reside within the data set. Therefore, for each possible parent there will be an amount of certainty associated with them being the parent of the child.

Working from these basic requirements and understandings we can start to define data structures which are sufficiently expressive.

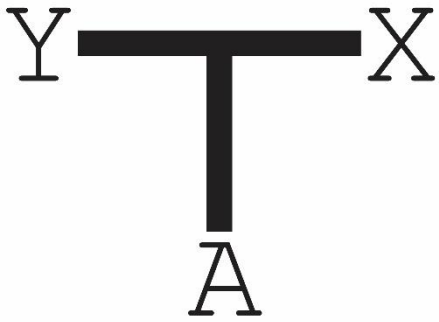


Figure 6 – A simple family tree.

A family tree can be seen in figure 6 which will act as a starting case study. As we look at figure 6, we can see that an object in our structure that is able to join together two parents and a child may be useful. We will see this basis feed into all the subsequent diagrams in this section. Figure 6 can be seen to represent the given structure under the initial interface of a child with a parental partnership which in turn has a male and female member. If we are then to extend this to suppose that rather than a single father and mother, but rather a set of males {B, C} and a set of females {D, E, F}, as shown in figure 7, the initial interfaces are no longer able to express this.

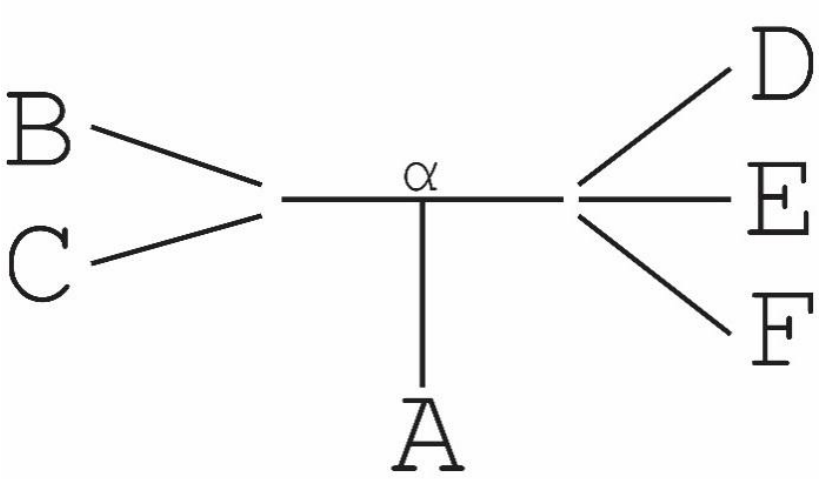


Figure 7 – A data structure with multiple possible mothers and fathers.

If we now consider the structure as laid out in figure 7, we see a structure where A is the child of the partnership α to which two possible fathers are connected and three possible mothers. This structure allows the idea of pairing between parents, as well as a concept of parenthood, which is not permitted by a more free form structure as seen in figure 8, in which links exist solely between individuals.

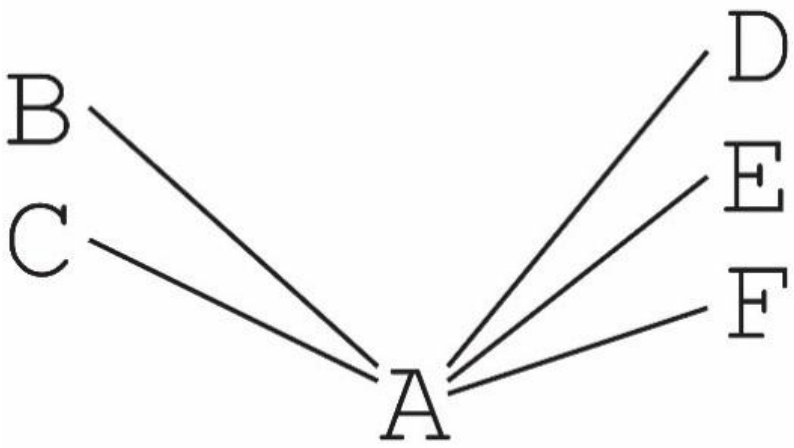


Figure 8 – An alternative structure for representing multiple possible parents.

The introduction of edges[3] into the structures means that information can be attached to these edges that denotes the provenance for the edge between the two connected entities. It may make sense to add an edge between the bottom of the partnership α and the child A in figure 7, both to allow for information to reside here but also to maintain consistency in the structuring, this will be further discussed in the next section.

[3] The term edge throughout is borrowed from graph theory and is used to refer to the lines connecting objects and people, which have also been termed as links in this paper.

4.5.2 Child, parents, sibling (Same parent sets for both siblings)

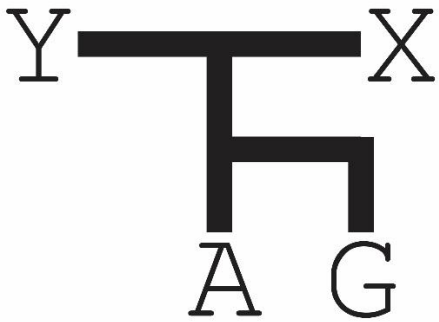


Figure 9 – A family tree depicting two parents with two children.

The next case study to be considered contains multiple children of a single set of parents as depicted by the family tree in figure 9. The initial thought here is to add multiple edges to the bottom of partnership α as seen in figure 10. This represents the fact that G has the same parents as A. However, given that we do not wish to make assumptions regarding the underlying data available to us, it is possible that the set of fathers and mothers for A and G will not be identical.

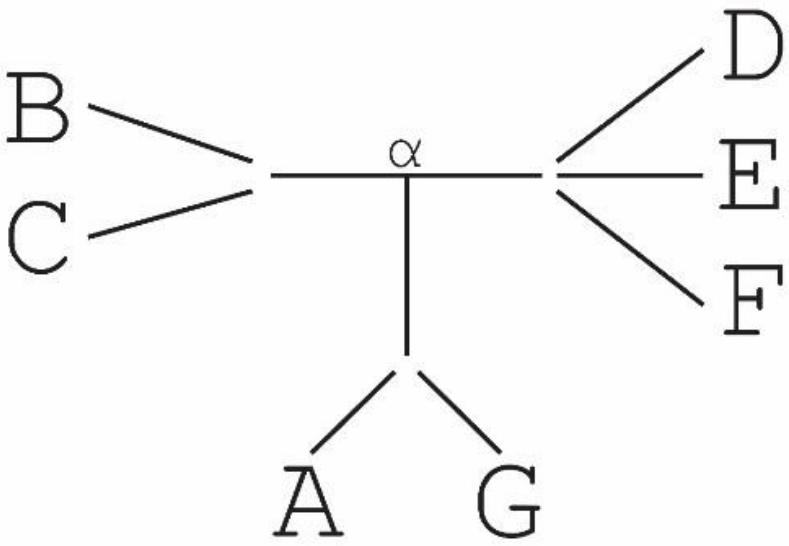


Figure 10 – A data structure representing two siblings with a set of possible mothers and fathers.

This stems from the complexities of the linkage processes and means it is likely that the sets of parents will be subsets, supersets or intersects of one another. If we extend this into a new case study the issue can be made clearer.

4.5.3 Child, parents, sibling (Parent sets vary between siblings)

The next case study shown in figure 11 builds upon the previous structure but introduces the stipulation that A's set of fathers is {B, C} thus intersecting G's set of fathers which are {B, H}. This can be seen in the colourings present in figure 11 and the description given in the figures caption.

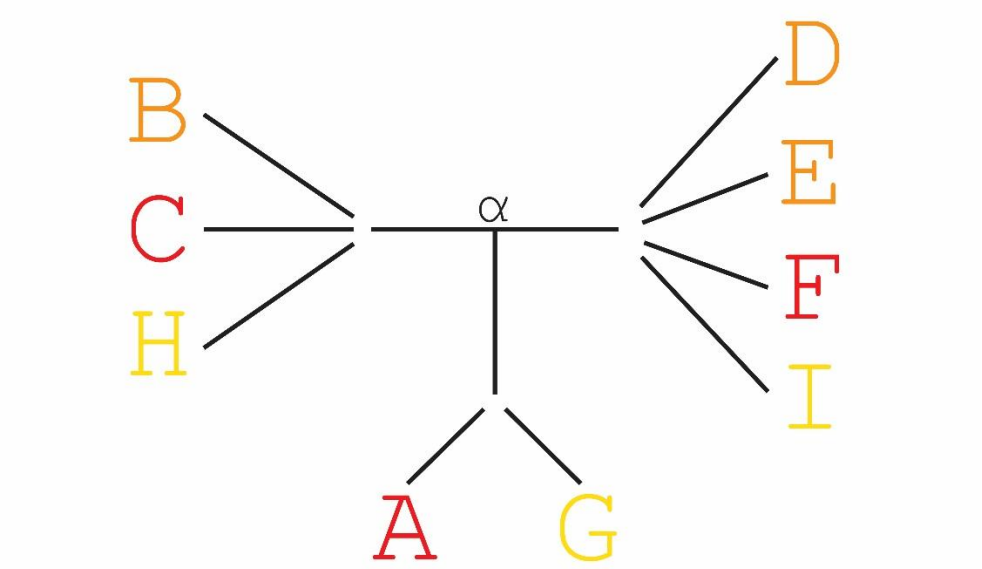


Figure 11 – A data structure using colouring to show the new assumptions we are making about the parents of each individual. The parents coloured that same colour as the child are possible parents of that child only, while parents of the colour orange are possible parents for both children. Therefore the possible parents of A are {B, C, D, E, F} and the possible parents of G are {B, H, D, E, I}

However, if we now remove the assumption that A and G are full siblings and that they could be half siblings, or unrelated, we introduce the possibility that the parental partnerships of A and G are not equal and that the current structure is unable to express the desired pedigree. The only time we can make use of this structure is where we can be certain of A and G sharing the same parents. We cannot assert this without a more complete knowledge of our data, which we are not able to assume. Therefore we need to further generalise the structure to express this.

If we consider figure 12, we can see a more generalised way of expressing a possible sibling relationship between A and G. This allows for a wider set of possibilities to be presented in the given structure. This includes the possibility that A and G both have an entirely distinct pair of parents (e.g. {A: C, F}, {G: H, I}), that they are half siblings either by a shared mother (e.g. {A: C, D}, {G: H, D}) or a shared father (e.g. {A: B, F}, {G: B, I}), or that they are full siblings (e.g. {A: B, E}, {G: B, E}).

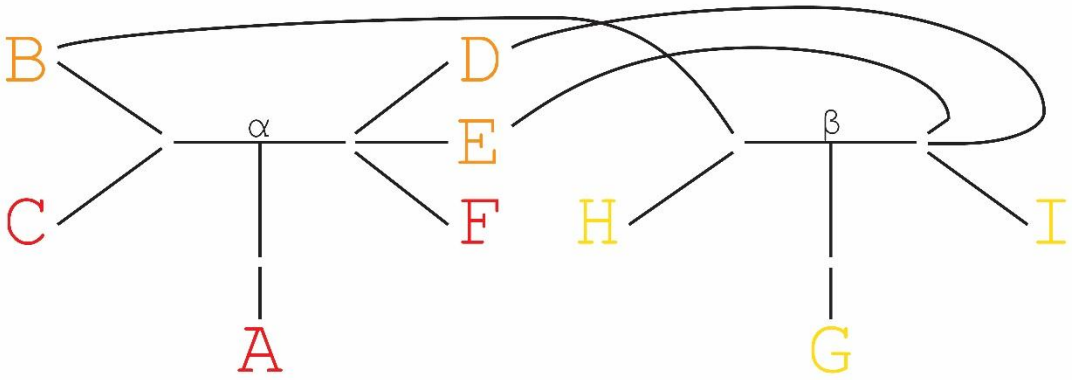


Figure 12 – A data structure of another approach to representing the linkage output. In this structure the idea of having a childbearing partnership object for each child is explored.

Next if we reinstate the stipulation that A are G are siblings we can consider how well the structure is now able to represent this. We cannot return to the structure as seen in figure 11 due to the need to maintain the additional information pertaining to the possible discrete parent linkages and especially in the light of uncertainty, as will be further discussed at a later point.

If we introduce a sibling bridge, as can be seen in figure 13, we can infer that A and G are siblings, and from that we can build a picture with greater certainty of the likely topology of the structure.

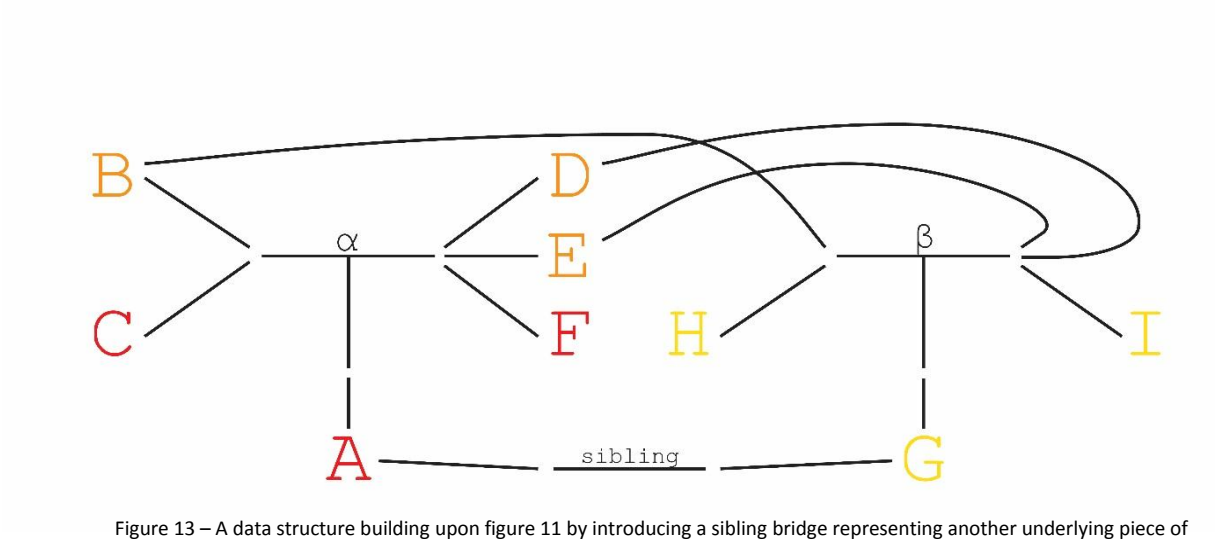


Figure 13 – A data structure building upon figure 11 by introducing a sibling bridge representing another underlying piece of information.

It is also worth making a side note about siblings and certainty without definite knowledge from census or similar data. This is not something that would be present in the current Digitising Scotland data set but as stated at the outset, the designed structure aims to be general enough to express any logical possibility and thus not be dependent on representing linked data only arising from a certain specification of source record. An example of a real source record from which a sibling bridge could be inferred would be in family census records. Census records allow us to see sibling relationships in a different way to using multiple birth records. For example, in the case of a census record, a household will be listed and the children in the schedule will be identifiable as siblings. However, in the case of birth records we need to find a child's parents and then find other children of that parent to identify siblings. Therefore, if we can find multiple ways to find the same genealogical relation we can have greater certainty that the genealogical relations occurs in the real world data set.

Additionally, the people linked to a sibling object are reliant upon linkage and so uncertainty exists, meaning that multiple people may emerge from either end of a sibling object. This will be explored with a more complex example in the case study see in figure 14. Before discussing this further, however, it will be useful to consider the edges in the graph and the annotations that will be found upon them if the structure is to fulfil its requirements.

4.6 Uncertainty

Now that we have created a structure with a reasonable degree of expression and started to put in place approaches to constrain its permutations by introducing further logical information, it may be worthwhile to discuss the uncertainty within the structure. Following this we will move on to introduce further structural components and consider how the structure scales, both computationally and conceptually.

As can be seen in figure 13, there are multiple edges connected to the right connection point of the object α, these represent three possible mothers of A, but we know that in reality it is only possible

for one of the persons D, E and F to be the mother of A. However, given the inherent uncertainty within linkage, our structures are not intended to make a definitive indication of a single edge, but instead look to present a range of edges. The usability of the given edges will nevertheless will benefit from having an estimate indicating which edges are more likely and which are less.

The root of any linkage lies in the source records of the original data sets. Therefore from an informed human perspective, it is important that these source records which give provenance to a linkage are stored in each link between parts of the data structure (i.e. on the edges). The underlying certainty of a given edge will either need to be provided by the linkage process, or it must be possible to recreate the certainty estimate from the provided provenance records. The use of these estimates when making queries to retrieve information in data structure are discussed in section 4.10. The placing of information on edges will require some significant changes to the initial interfaces which are discussed in section 4.9.

It is also important to note that we are unsure of how accurate or representative a certainty estimation it will be possible to generate, in the idealised linkage process. This also extends to the use of those estimates subsequently in queries to calculate overall certainty of a query. Therefore, it is important to use these estimates carefully, limiting their use to using them as comparisons between a set of result objects from a single query, rather than to compare two distinctly different links in the structure.

4.7 One-to-One Object Enforcement

Another important concept within the defined structure is of one-to-one object enforcement. We assume that the idealised underlying linkage process will enforce this constraint on the produced set of linkages. This assumption is based upon a set of absolute truths that are defined by the nature of all source records. Take, for example, a birth record: there is no more than one for each person detailed in the data set, we can, therefore, enforce that only one person be assigned to the base of an intermediary partnership object and then the attachments to its mother and father points being the possible sets of parents in relation to the linked child. This enforcement allows for a structure which can be addressed from a single person and the uncertainty is encapsulated in the 'sideward steps' (i.e. the edges between partnership member points (e.g. mother, father, husband, wife, sibling) and persons) in the structure. This data structure gives sufficient expression for the logical genealogical possibilities and there is therefore no reason to allow the uncertainty in the structure to spread beyond and also appear between children and childbirth objects; as we know that there is one childbearing partnership for each child.

The sibling bridges that exist in the structure also will need to be subject to the same one-to-one enforcement. If we consider a census record that details a family with 3 children and we denoted the groupings of possible individuals for each child by the identifiers {A, B, C}, {E, F, G, H} and {D, E}. We can logically deduce that each possible individual has two siblings and so will be attached to two sibling bridges, as shown in figure 13. By limiting the number of sibling bridges to the number of sibling possibilities presented in the paper records we can be sure that every sibling bridge in the structure represents an actual genealogical relationship. This is opposed to an approach of creating a sibling bridge for each possible permutation of sibling relationships between those found in the sets three given sets. This would lead to a need to attach a certainty estimate to each bridge in order to decide which are most likely to be true, and on top of this a mechanism to prevent the creation of more sibling relationships than records exist for. In the suggested bridge structures the uncertainty estimates are limited to the edges between the intermediary objects and the individuals (the calculation of an overall certainty estimate will be discussed in the querying approach) but also inherently enforces that only one individual can be the correct attachment to an end of a bridge removing the need for an additional mechanism.

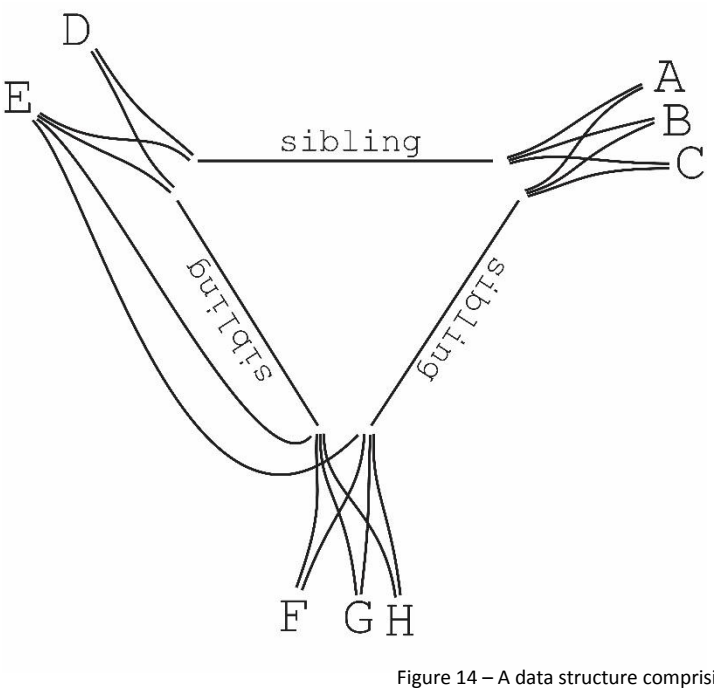


Figure 14 – A data structure comprising only of sibling bridges.

An issue now arising from our bridge approach is of the same individual being attached to both ends of a sibling bridge. From a linkage viewpoint we could expect this to happen if two siblings both have the same forename resulting in the census records likely being overlaid on both birth records of the same name. Logically we expect that the linkage algorithm will not make such a suggested linkage as it will be able to identify that it is attempting to identify a person as its own sibling. Given this, we assume that this will not be an issue for our representations. However, if it is not possible to assume such behaviour from the underlying linkage process then we will need to do further work to identify ways of restructuring sibling bridges in situ to remove the self-linkage while maintaining the remaining set of suggested possible siblings.

4.8 Extending Sibling Bridges Concept to Marriages

We have now laid out a way to insert intermediary objects into the structure that make bridges between individuals that represent a collection of genealogical relationships. In the sibling case this collection is the unification of two parent-child relationships. The sibling bridges can exist in two environments: either alone, without the common parent being identified (as could be derived in figure 14) or alongside other intermediary objects which therefore allow for two possible 'paths' across the structure. Paths represent a way in which we can traverse the data structure from one person to another; the steps taken in this path indicate the relationship between the two people. It would be reasonable to expect that the presence of two paths (arising from different source records) across the structure, both giving the same conclusion, increases the certainty of the underlying genealogical relationship. It would be possible to see either path being a supporting path to the other, but given the sibling bridge makes a bridge across the structure, we would prefer to see this as the supporting path to the genealogical step by step path (i.e. child to parent to other child - who is the sibling); although in the presence of only a sibling bridge we can still make use of the bridge to build genealogical structures. This concept of using additional source records as bridges within the structure to increase certainty and reasoning can be extended to other sets of source records.

If we consider marriage records, we can explore how these can be used to act as supporting records in a similar vein to that of sibling bridges. In the case of marriage records, they attest to social relationship structures rather than genealogical relationships, but due to cultural expectations we can make genealogical relationship decisions due to their close relatedness. For example adoption, wills,

life insurance and next of kin, could also be used as indicators of genealogical relationships even though in themselves they are only indicators of social constructions.

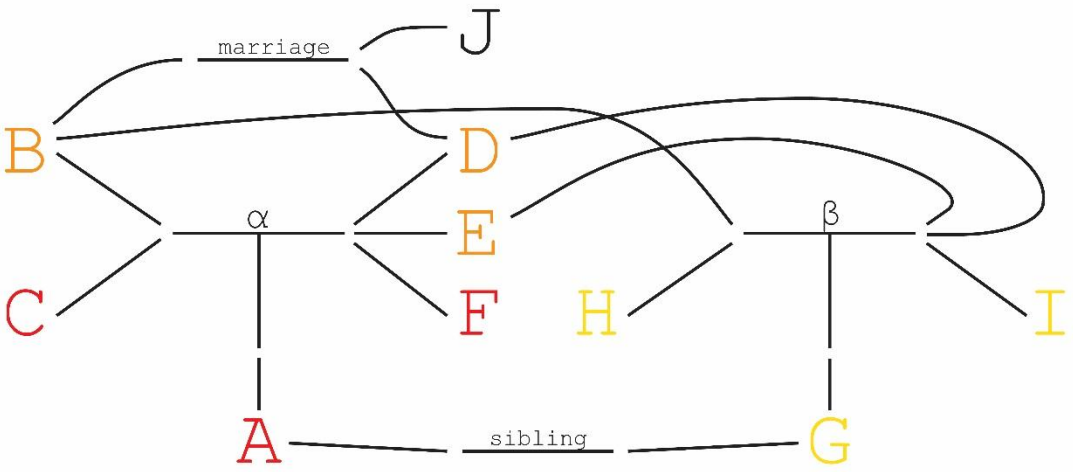


Figure 15 – A data structure using a marriage bridge to represent further information.

If we take figure 15, we can consider the value of bridges in the data structure that derive from social constructs rather than genealogical descent. A marriage bridge has been added into the structure which shows individuals D and J as a possible spouse of B. J has been added into the example to demonstrate that multiple individuals can be attached to either end of a marriage bridge, however their sex must be representative of the correct end of the bridge. This is the case for all diagrams with the left being male, in this case the husband, and the right being female, in this case the wife. However, if we suppose that the certainty estimates on the edges attaching B and D to the bridge are the most significant then we can begin to deduce the effect that the information in this bridge implies for the wider structure. Given more traditional social norms, it would be expected in a reasonable number of cases that the parents of a child will at some point be married and so a marriage record will exist. The record of the marriage gives rise to the bridge in the structure and would lead to a natural assumption that it is more likely that B and D have produced children together, thus increasing the likelihood that A and G are children of B and D.

At this point of query it will be important to define a formula that is able to combine the certainty estimates of a set of edges within the query area and produce a combined certainty estimate representing the most likely 'bigger picture' pedigrees from the structure. This combination mechanic when dealing with social indicators of genealogy will need to make consideration of the change of such influences on certainty estimate over time. A consideration of geographical location and individual indicators (i.e. religion, political affiliations, occupation, bumper stickers) could act as a proxy of cultural influence on individuals.

In the case of figure 15, we could also see how, if the marriage bridge created a link between C and F instead, it would increase the likelihood of individuals C and F being the parents of A and leave the parents of G to be unaffected by the marriage bridge and to be handled separately. In this case it is better to suppose such a pedigree in the absence of the sibling bridge, due to the possibility that the sibling bridge is incorrectly linked.

One idea considered in the design of this model was the possible flattening of the marriage bridge on top of the childbearing partnership α. Under the structure proposed in figure 10 this would be a possibility, however in the decision to place each sibling into separate childbearing partnerships we prevent ourselves from being able to flatten the bridge across all the partnerships as doing so would limit the set of possibilities we can express using the model. A step away from the need for the defined structure to remain flexible and preserve the uncertainties produced in the linkage process. To do

otherwise would result in both the loss of information, resulting in less informed queries and make the application of a marriage bridge counterproductive. For example in the case of figure 15 if we are to flatten the marriage bridge upon α we make a further stipulation that the members of the marriage record must also be the parents of A – if we are to support uncertain data we will not be able to say this. A possible refutation of such would be to consider that in our structure B and D are a married couple but the actual parents of A are C and E. If we had enforced the flattening of the marriage bridge onto the childbearing partnership then it would not be possible for this pedigree to be suggested. The underlying issue here is that by the flattening of the marriage bridge onto the childbearing partnership we are implying that the genealogical action is a certain influencer of the social construction which is an assumption that we know not to be correct.

A noteworthy point from the idea of flattening the marriage bridge onto the childbearing partnership is the value of using bridges as supporting records to the partnership linkages and also the possibility of considering multiple bridges that could pertain to the partnership. A probable approach to doing this would be to traverse the structure to find bridges attached to the individuals who in turn are attached the considered partnership, rather than placing the details of the associated bridges on the partnership object.

The structure that has now been defined allows for us to create expressive genealogical structures which are able to be constructed with an appreciation of uncertainty arising from the underlying linkage process. We have explored the use of genealogical based relations as the central element of the structure, although have remained source record agnostic throughout. The structure is also able to make use of social constructions that have genealogical implications to provide support in making decisions. In the given example we have considered the case of marriage but further research could look at considering other social and cultural based concepts that support genealogical relationships.

4.9 End Point – Linked Interfaces

The data structure that we are developing throughout this work is underwritten by interfaces and can be seen in figure 16.

The initial interfaces have been adapted in a number of ways. The way in which they join to each other has been redefined introducing Links between them rather than single integers to reference one another. Also the old IPartnership interface has how been split into two versions, one for childbearing relationships and one for marriage relationships. This means there is no longer a need for marriage to be stipulated because of childbearing or vice versa. In the end it was decided not to fold the interfaces back into the original interfaces, as the structures we are now representing, due to their inherent uncertainty, are too far removed from the other models for them to be presented by the same interface. The structuring of a Link is also detailed here to facilitate a better understanding of the data structure, even though it is not an interface.

```java
public interface ILinkedPerson {

    int getId();

    String getFirstName();

    String getSurname();

    char getSex();

    Date getBirthDate();

    String getBirthPlace();

    Date getDeathDate();

    String getDeathPlace();

    String getDeathCause();

    String getOccupation();

    List<Link> getChildBearingPartnerships();

    Link getParentsPartnershipLink();

}
```

```java
public interface ILinkedMarriagePartnership {

    int getId();

    Link[] getPerson1PotentialLinks();

    Link[] getPerson2PotentialLinks();

    Date getMarriageDate();

    String getMarriagePlace();

}
```

```java
public interface ILinkedChildbearingPartnership
    extends Comparable<ILinkedChildbearingPartnership> {

    int getId();

    Link[] getPerson1PotentialLinks();

    Link[] getPerson2PotentialLinks();

    Link getChildLink();
}
```

```java
public class Link implements Comparable<Link> {

    private Evidence[] provenance;
    private float certaintyEstimateOfLink;

    private LinkedPerson linkedPerson;
    private IntermediaryLinkObject intermediaryLinkedObject;

}
```

Figure 16 – The new Linked Interfaces, and the Link class which is presented here to facilitate a better understanding of the given interfaces.

4.10 Queries

Next we will consider approaches to querying and identifying localised relationships ordered by certainty estimates from the defined structure. A query approach is needed for the new interfaces due to the uncertainty they contain, this is opposed to the APIs for the original interfaces which returned a single result, which could be described in linkage terms as 'best fit'.

Before considering how to query the structure it is useful to outline the queries we need to be able to make. In this paper we will restrict ourselves to considering parental, child, sibling, and marriage queries. These act as the building blocks for traversing any genealogical structure and more complex queries can be built using these. The optimisation of the combination of these initial queries will also be briefly discussed but represents an interesting area for further research which pertains more to the complexity than the expressive power of the querying approach.

4.10.1 Parent Queries

The parent query is summarised as: **Given person A, find the possible set of mothers or fathers.**

The query approach is much the same whether or not we are looking for possible fathers or mothers, therefore this following paragraph talks about finding fathers but can be applied equally to mothers.

The structure that we have defined details that to find the father of a person we must first identify the set of childbearing partnerships to which the child is attached. Identifying and traveling along this edge is a simple process due to the one-to-one enforcement of birth records to data structure objects meaning each person is attached to at most one childbearing partnership. As the relationship between the child and the partnership object is one-to-one the certainty estimate of the edge between the two can be asserted to be 1 (i.e. without doubt). From the partnership object we can then consider all of the attached males. Each will have associated supporting evidence records and a certainty estimate as output by the linkage process (or calculable from the given evidence records). The combination of the certainty estimates for each will be a simple multiplication with the initial edge. If a father can be linked by alternative paths back to the partnership object, they will be weighted more favourably.

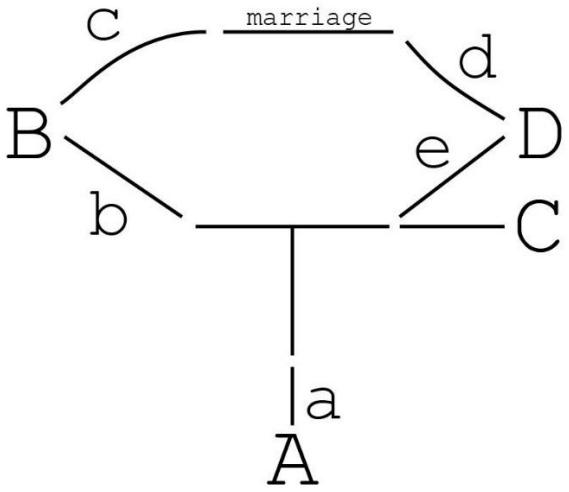


Figure 17 – A data structure with edges labelled in corresponding to the specified formula.

Considering figure 17, such a weighting will be calculated by the formula (where h is the certainty estimate for each given edge):

$a.h \times b.h + F_m(c.h \times d.h \times e.h) + \ldots$

The factor F_m is used to scale the impact of the additional weighting. This will comprise an element of the significance of a marriage bridge based on the rate of occurrence in the structure compared to the expected rate for the original population (i.e. the number of marriages found in the data structure compared to the number we expect to find in the real world population) and another element pertaining to a cultural approximation of the significance of marriage as an indicator of genealogical relationships in the child's year of birth.

By considering each possible father in this way, a set of fathers will be identified with associated combined certainty estimates. By ordering these, it will be possible to identify the list of most likely fathers.

The process is the same in the case of mothers.

4.10.2 Child Queries

The child query is summarised as: **Given a father, or a mother , B find the possible set of children.**

The approach to the child query is similar to that of the parent query. The same traversal across the structure is being made but in the opposite direction. The same idea of using social constructions such as marriage bridges to increase the certainty estimation of the child with whom the father has an alternative linkage path can be used to give a greater certainty estimate weighting to a more likely child. In the child case it is also possible to extend the alternative paths to consider sibling bridges as

well, for example using a premise that my sibling's father is likely to be my father too. This is laid out in figure 18 with the combined estimate being calculated using the formula:

a.h x b.h + F_m(c.h x d.h x e.h) + ... + F_s(f.h x g.h x h.h x i.h) + ...

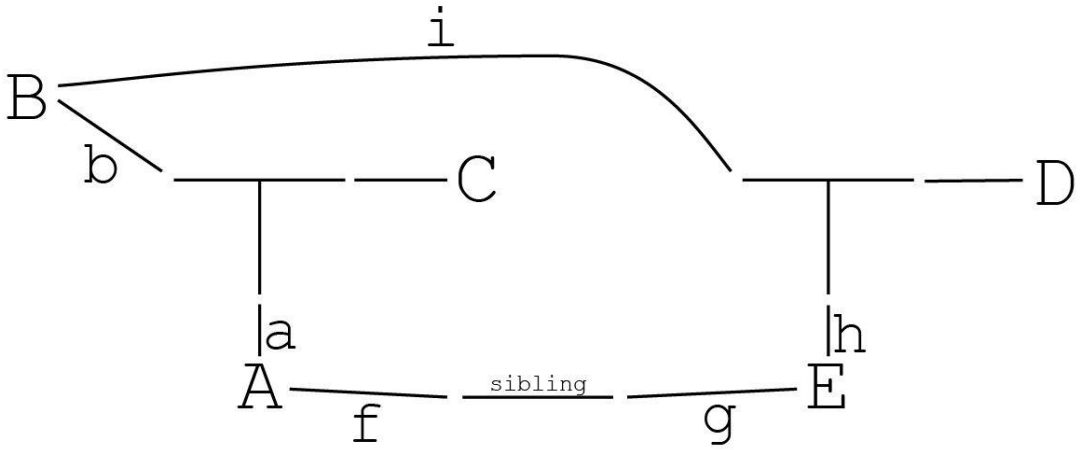


Figure 18 - A data structure with edges labelled in corresponding to the specified formula.

The factor F_s is again a scaling factor and will add an impact estimate weighting that comprises considering the number of sibling bridges in the structure compared to the expected number of bridges. This is based on an estimation of the number of sibling bridges that could exist in the population from the size of the population and the average number of children per family for the given setting, as defined by the below formula:

$$\text{Approximated expected number of sibling bridges in population} =$$

$$\frac{((\text{average children} - 2)^2 + \text{average children} + 2) \times (\text{population size} - 1)}{2(\text{average children} + 1)}$$

$$\{\text{average children} \in \mathrm{R} \mid \text{average children} > 2\}$$

No cultural approximation is involved in the factor as sibling bridges are based upon genealogical relationships rather than a social indicators of genealogical relationship.

The use of these factors would not be necessary if we believed that the supported records existed for all sibling relationships. However, given the patchy nature of genealogical data sets, being able to forecast how significant bridges that do exist are to the structure is important.

4.10.3 Sibling Queries

There are a number of nuances to sibling queries due to the varying degrees of relatedness that siblings can share.

4.10.3.1 Half Sibling Queries

The half sibling query is summarised as: **Given a person A, find the possible set of half sibling with either a common mother or father.**

The approach to half sibling queries will look at identifying the siblings arising from either the father or the mother. The query approach is much the same whether or not we are looking for possible fathers or mothers, therefore the following paragraph talks about finding siblings on the father's side, but can be applied equally to the mother's side.

The query will need to state the person for whom siblings are being found. The first step in the query will be to identify the partnership of which the person is the child. From here the set of fathers can be considered. Each possible father will be linked to a number of other childbearing partnerships and each of these will have an attached child who is a possible sibling on the father's side of the initial child. Given the additional traversal steps compared to the other queries, a larger number of results are likely to be found, but this is to be expected given the branching nature of genealogical structures. Each of the identified possible siblings will need to have a combined estimate calculated. The combined estimate in this case is calculated using the following formula in conjunction with the edge labels seen in figure 19:

a.h x b.h + F_s x MAX(a.h x c.h x d.h x b.h, a.h x e.h x f.h x b.h) + ... + F_m(i.h x j.h) + ... + F_s(g.h x h.h) + ...

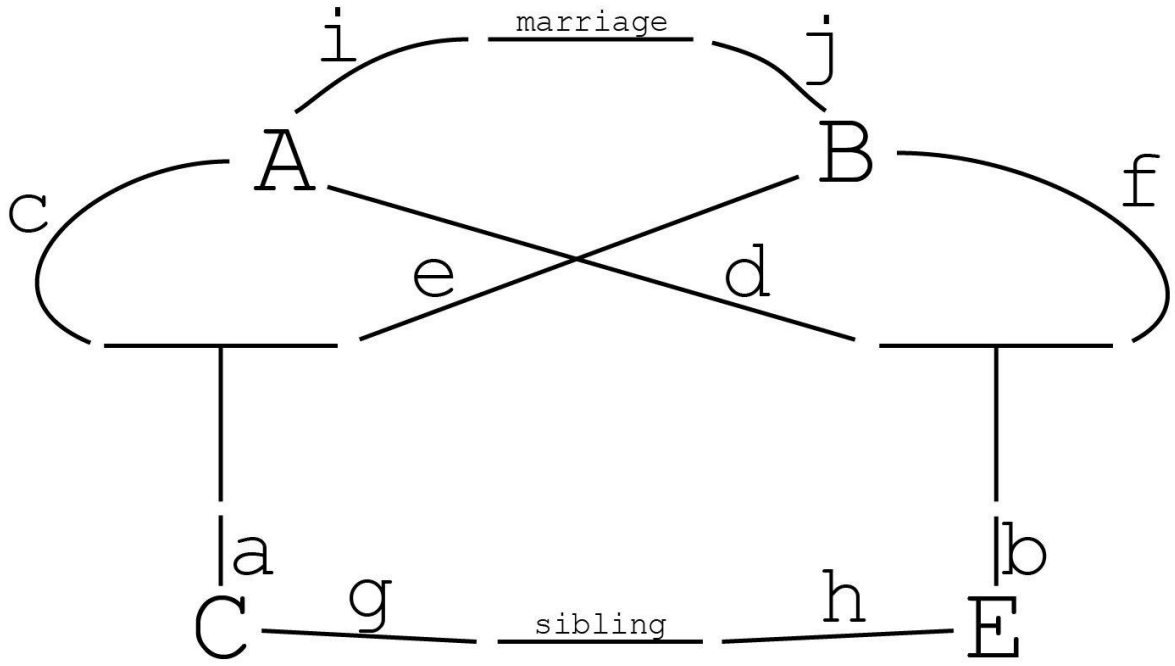


Figure 19 - A data structure with edges labelled in corresponding to the specified formula.

He main constituent of the estimate is made up of the combination of the child edges and the maximum intermediary paths between the two. Some weighting is also given if the common parents share a marriage bridge.

4.10.3.2 Full Sibling Queries

The full sibling query is summarised as: **Given a person A, find the possible set of full siblings where both the mother and the father are the same.**

The approach to the full sibling query is to identify the union of the half sibling results for siblings on both the mother's and father's side. The combined estimate in this case is calculated using the following formula in conjunction with the edge labels seen in figure 20:

(((a.h x b.h x c.h x d.h) + (a.h x e.h x f.h x d.h)) / 2) + F_m(i.h x j.h) + ... + F_s(g.h x h.h) + ...

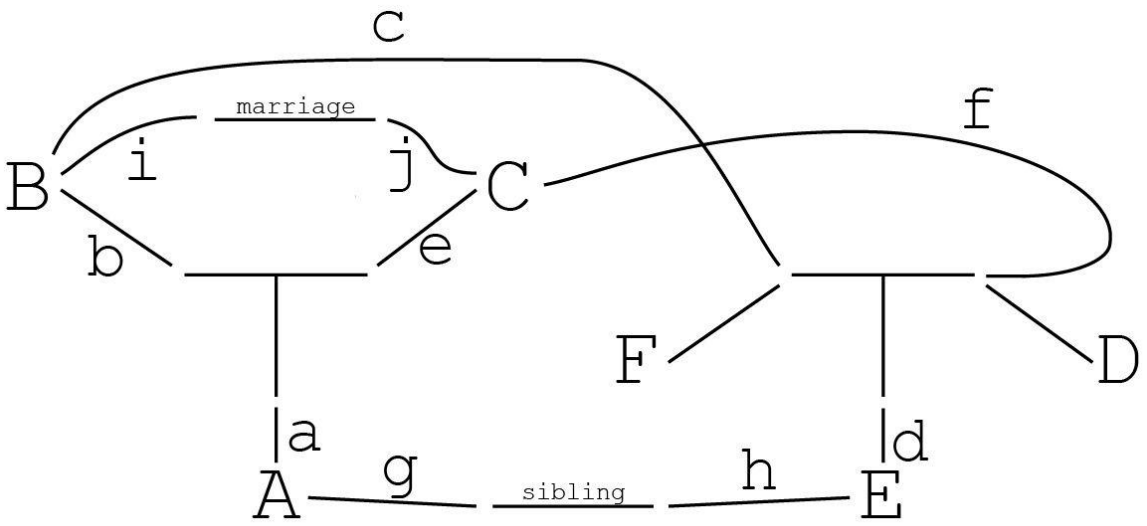


Figure 20 - A data structure with edges labelled in corresponding to the specified formula.

Here the main constituent of the estimate is made up of the average of the parent combined edges and intermediary paths with the weighting factor consisting of any marriage bridges linking the two parents and also any sibling bridges linking the two children.

4.10.3.3 Sibling Bridge Queries

The parent query is summarised as: **Given a person A, find the possible set of siblings, either half or full, to whom person A is connected to by a sibling bridge.**

The approach to the bridge sibling query given a person is to consider all the sibling bridges of the individual outwith of the content of direct genealogical relations. Therefore this method will be more useful in addressing siblings whose common parent falls out with the data set or is unidentified. The certainty estimate in this case is calculated by use of the following formulas in conjunction with the edge labels seen in figure 21:

In the half sibling bridge case:

$a.h \times b.h + F_s \times MAX(g.h \times c.h \times d.h \times h.h,\ g.h \times e.h \times f.h \times h.h) + \ldots$

In the full sibling bridge case:

$a.h \times b.h + F_s(g.h \times c.h \times d.h \times h.h) + \ldots + F_s(g.h \times e.h \times f.h \times h.h) + \ldots + F_m(i.h \times j.h) + \ldots$

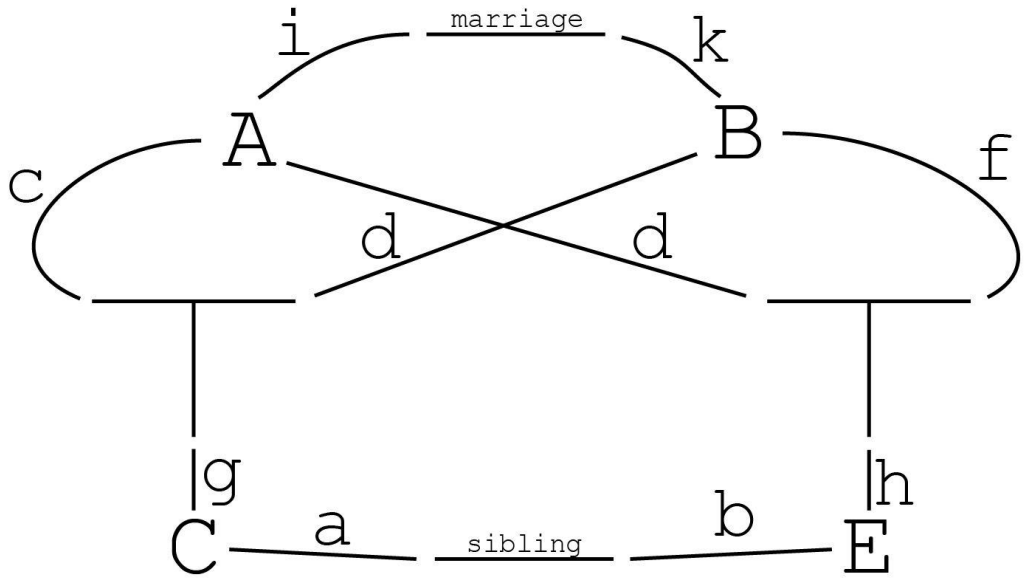


Figure 21 - A data structure with edges labelled in corresponding to the specified formula.

Where the sibling bridge indicates that they are half siblings, then the main component of the estimate is made up of the edges attaching the two individuals to the sibling bridge. The weighting factor increases the estimate by the maximum likely alternative path combined certainty estimate.

In the full sibling cases, the main component again is made up of the edges attached to the two individuals to the sibling bridge. The weighting factor comprises the combined alternative sibling paths with scaling and the paths of any marriages containing the two common parents.

4.10.4 Childbearing Partner Queries

The parent query is summarised as: **Given a person A, find the possible set of people with whom they have produced a child.**

The approach of the childbearing partner query given a person is to first identify all childbearing partnerships to which the person is attached. From each of the attached partnership objects we can then consider all the other attached individuals of the opposite sex. Each of these is a possible childbearing partner for which a combined certainty estimate can be calculated using the following formula in conjunction with the edge labels seen in figure 22:

$$a.h \times b.h + F_m(b.h \times c.h \times d.h) + \ldots + F_c((a.h \times b.h \times e.h \times f.h + F_s(i.h \times h.h \times g.h \times j.h)) + \ldots$$

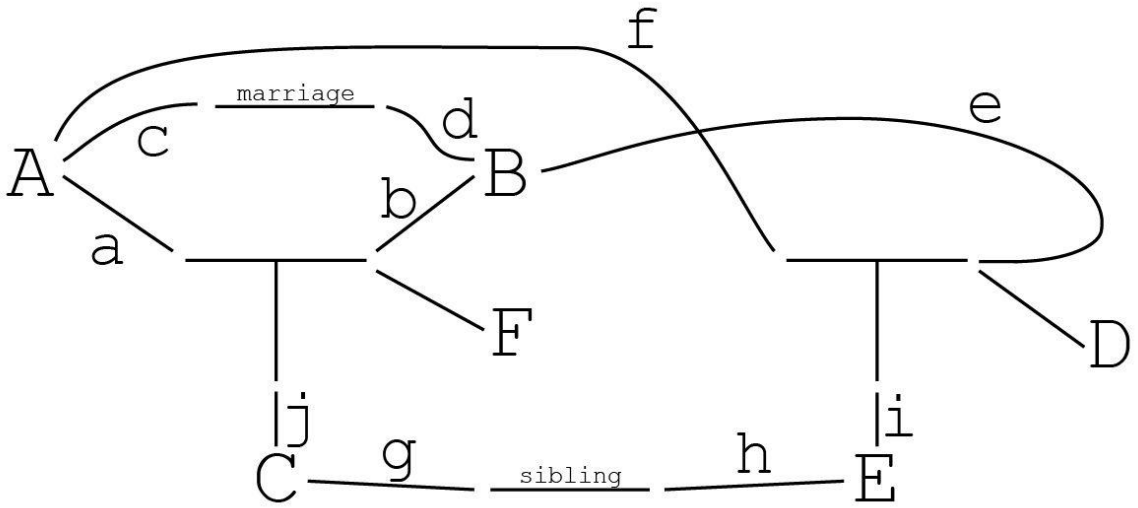


Figure 22 - A data structure with edges labelled in corresponding to the specified formula.

The factor F_c is used to scale the impact estimate of the additional weighting. This will comprise an element of the significance of there being multiple possible children born between the initial individual and the other individual. The factor F_c will make use of an approximated value for the representation of the supporting marriage bridges in the data set and gives a value to the relational monogamy observed in the real population, calculated by:

$$F_c = Scaling\ Factor\ \times \frac{2\ \times\ \text{marriage bridges}}{\text{population size}\ \times\ \text{proportion of real population married}}$$

4.10.5 Marriage Bridge Queries

The parent query is summarised as: **Given a person A, find the possible set of partners to whom person A is connected to by a sibling bridge.**

The marriage bridge query is of a similar ilk as the sibling bridge query which is again more record administrative focused. The approach taken is to consider all the marriage bridges of the given individual. Given the social construction of marriage the bridges when queried need to be seen as social indicators of genealogy rather than genealogical truths. However, such a query still has uses within a wider context of linkage when a focus is needed upon social and cultural structures, for example when researching the effects of social policy and legislation on a population. Once the set of marriage bridges has been established, the set of individuals attached to the other side of each of these can be considered, and a combined certainty estimate can be calculated for each using the following formula in conjunction with the edge labels seen in figure 23:

`a.h x b.h + `F_c`(c.h x d.h) + ...`

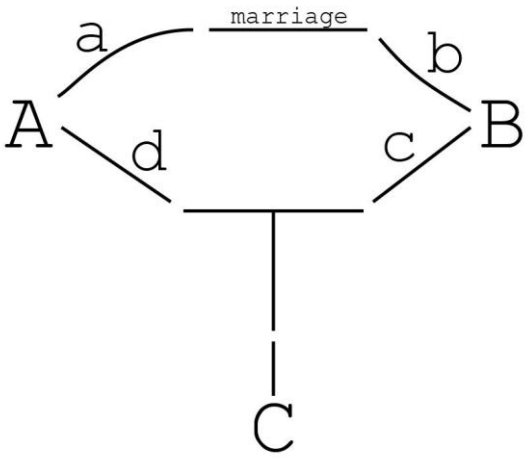


Figure 23 - A data structure with edges labelled in corresponding to the specified formula.

The additional weighting increases the certainty estimate of the marriage if childbearing partnerships between the two individuals exist. The weighting is scaled with a factor representing the recorded monogamy across the wider population.

4.10.6 Further Queries

There may be grounds to consider further queries as the need arises. These may look at making larger step traverses across the population, for example in the process of identifying the set of X degree cousins of an individual. Also in the presence of additional social based bridges within the structure it would be possible to factor these into the scaled weighting values.

It is useful to discuss the need for the combined certainty estimate calculations within a representation model of linked data due to this being different due to the introduction of uncertainty. Traditional approaches to linked data look to make sets of potential linkages and then run algorithms to create a best fit set of links that are then returned. However, the hypothetical linkage process we consider here does not seek to return a single best fit linkage and instead returns a set of uncertain probabilistic linkage possibilities. Therefore, while multi-step (i.e. siblings, cousins) queries are inherently presented as certain within a best fit linkage, they cannot be with this idealised linkage approach, when uncertainty is exposed in its results. However we still need to be able to construct multi-step queries in our data set and therefore a way of evaluating the combined certainty estimate of the given query has to be defined. This estimate, as demonstrated in above formulas, draws on the certainty estimations of the traversed edges and also considers cycles in the graphs supporting the same relationship by an alternative path, to add more weighting to the people supported by these cycles. The presence of this concept at the representation level when querying the data set appears to be necessary due to the initial outlining of the idealised linkage algorithm, laid out in the introduction. Future work on the idealised linkage algorithm will need to consider ways that the estimate combination formulas can be encapsulated into the linkage process. This may require a reconsideration of where the line between linkage and representation is drawn when designing linkage algorithms that expose uncertainty.

4.11 Textual Justifications

The information returned for each query will need to be able to fully represent the uncertainty in the structure across the multiple possible results for each query. The results set could be expected to return a list of identifiers and coded values in response to the query that can then be interpreted by a trained user. However, this step of training serves as a stumbling block to a wide range of users being able to make use of the representation system, and by extension linked data, with uncertainty.

Therefore, time has been given over to look at approaches of expressing query results in natural explanatory language.

The information we want to return at the point of query includes: the type of query, the possible individuals that have been identified, the combined certainty estimates, the supporting bridges and the underlying source records that support the traversed edges in the data structure.

Therefore, a summarised way of textually representing this data may take on a form, for a sibling query, such as:
`The most likely father side sibling with a certainty estimate of <0.X> is person <person> with <person> as the common parent supported by the evidence in records A, B and C.`

Some amount of language tailoring through each subsequent line may also be useful to set each line in its surrounding context but only to a limited extent as to vary language style too much may lead to a lack of a consistent framework, which is good to maintain across texts pertaining to similar entities. Also considering how to best present the certainty estimate in the textual justification is important. To present them as a numeric value allows for a better technical understanding of the returned data. However, to present this textually may make it easier to understand for non-technical users and also may offer the opportunity to choose the positive strength of the certainty estimate indicator word in light of the strength of nearby links as opposed to using a predefined scale. The explanation of the way in which the source records and their associated parts support the linkage will also need further consideration at a later point. In order to support this we would require the implementation of the evidence module to be specific to the source record structure. This raises a number of issues due to the number and variation of record types and also needing a better understanding of how the idealised linkage process is performing its linkage, which is beyond the remit of this work.

However, it is still important to offer a generalised evidence structure whose integration with an underlying linkage could enable the aforementioned information to be presented at a point in the future. The discussion of whether such evidence information, including indications of the supporting part of the record, would be stored and output from the linkage process or would be calculable post linkage, given an abstracted set of the linkage algorithms will require further consideration. However, in the event of a post linkage approach being taken further discussion of the splitting point between linkage and representation in the idealised structure will need to occur, as has already been mention in relation to the combined certainty estimates.

4.12 Query Language

In this section the conceptual ideas behind a query language for the defined data structure is discussed, although it should be noted that an implementation of this has not been made within this project.

If linked data is to see widespread usage then it will require that the use and availability of both suitable data sets and also tools to perform linkage are made more widely available. The former of these is based on a number of elements but notably public support. An element of making linkage systems that are available and easy to use points to a need to offer a way to interface with linked data sets that does not require a high level of technical understanding. The above discussion of textual justifications has a role to play as well as creating a way for people to form queries that is abstracted away from the code base.

A query language for interacting with uncertain data would be a good approach to doing this. The query language would need to offer the functionality to make the block queries available to the user,

these being father, mother, child, childbearing partner, marriage partner, half sibling, full sibling, sibling bridge.

It will also need to offer the ability to link these queries together to allow wider traversals of the structure to be made, in which case it will be beneficial to offer loops within the language. Loops would be especially helpful in the forming of ancestral line queries. It would also be useful for the language to offer the ability to set a minimum certainty estimate to be considered significant enough to make returning a result worthwhile. Functionality to set this as a global parameter and also at the point of query would be useful. Based upon the ideas explored of using social and cultural values in the weighting elements of the certainty estimate providing the functionality to adjust and override these within the query language may also prove useful, although the control of these via a config file (or as a global parameter) may be a more standardised way of holding these values given the change in them over time based upon dates found in queried records. Furthermore offering flags to be used with queries to choose if a textual justification or a numerical based value should be outputted in response to queries will allow a choice of usage.

5 The Implementation

In this section we will talk about the details of the implementation of the data structure.

The code base can be found in the package /population_model/population_representations on the repository found at the address:
http://quicksilver.hg.cs.st-andrews.ac.uk/digitising_scotland

The Continuous Integration server for the project can be found at the address:
https://builds.cs.st-andrews.ac.uk/job/digitising_scotland/

Further information about the package and installation can be found at the address:
http://digitisingscotland.cs.st-andrews.ac.uk/population_model/index.html

As mentioned earlier a set of interfaces were used as the starting point of the structure, however these have seen considerable changes to reflect the different structuring of a link based population and also due to the need to break apart the IPartnership interface which implied that children have to be born out of marriages. However, at the conclusion of this section we will look at structuring a new set of interfaces which enable backwards compatibility with the old interfaces while still supporting the new structures that we have created in the linked population.

5.1 Linked Persons

The linked person is a slight adaptation on the person interfaces found within the other population models within the wider project (i.e. Organic) which implements the original version of the IPerson class. The changes made were to make use of arrays of links to join together people and objects, rather than the previous approach of using the ID of the object. The same extends to where lists of integers were originally returned. These adjustments allow for more information to be stored in the structure. These can be seen as annotations on the edges of the graph. Also each Person has a list of links which identifies possible childbearing partnerships, marriage bridges and sibling bridges which the individual may be a member of. These enable for the linked structure to be built with the people within the structure able to be linked to intermediary link objects in keeping with the underlying source records.

5.2 Intermediary Link Objects

The structures explored in the design section can be seen as graphs with a number of nodes and edges. The edges are links that will be discussed in the next section. Any of the nodes that are not people are extensions of the abstract class Intermediary Link objects. What the link object represents depends on the class instance that it has initialised. The decision to maintain the implementation approach of using objects in-between people, as found in the earlier population models, is partly due to the continued implementation of the IPopulation interface and also the centralised control of each of the objects in the population instance.

The abstract class offers two lists of links. Each list pertains to the set of individuals that are linked to either end of the object, each instance is given a unique ID and also a reference name, although this has only been added to make case studies easier to understand, which will be further discussed later.

5.3 Childbearing Partnerships

The childbearing partnership object extends the intermediary link object class and makes use of the two lists of links, one for possible fathers and the second for possible mothers. It also adds a link of its own to attach the child to the link object. The childbearing partnership is used to implement the T shaped objects seen through the earlier examples. Methods are also offered so that they may be accessed by the term of father or mother for ease of use and population constructions.

5.4 Bridges

The implementation of bridges in the structure also extends the intermediary link object making use of the two lists of links.

5.4.1 Marriage Bridges

Marriage bridges also implement the link adapted ILinkedMarriagePartnership interface which enforced that information pertaining to the marriage event is accessible. Marriage bridges also provide specialised methods to indicate the correct use of the lists.

5.4.2 Sibling Bridges

Sibling bridges also offer a pair of specialised methods to indicate correct queue usage and contains the sibling type enum to provide further detail about the nature of the represented sibling relationship.

5.5 Links

The Link class is the most important class in this model. The link class enables the storing of the supporting information at the places where it is most relevant. A link always joins together a person and an intermediary link object. A link also has an estimate for certainty and holds an array of evidence objects which indicate the provenance of the link based on the underlying source records. Link initialisation is self-referencing and uses the single constructor to attach the possible partners/parents/siblings to the correct intermediary link object. A second constructor is providing for linking a child to the base of the childbearing partnership.

5.6 Evidence

The evidence class provides information about the underlying source records and the part of the record which was notable in this case. The implementation here limits itself to identifying the record ID, type, and the part of the record supporting the given linkage. These are then used later in the textual justification to further demonstrate the ways of giving better explanations in data sets with uncertainty. The limited implementation of evidence, as noted above, is due to the need for a defining of the underlying linkage approach to evidence identification and output. Additionally the large number of record types that would need to be surveyed to be sure of creating a general class able to be used to communicate the full nuancing of the various record types extends beyond the time limitations of this dissertation.

5.7 Types

A number of Enum classes are used to organise and control sets of options in certain areas of the model.

5.7.1 Query Types

The query type enum details the full set of possible queries. It is used to identify the query type, at query time, at the return point for results, and also for textual justifications. The currently supported query types are:

```
CHILDREN
FATHERS
MOTHERS
CB_PARTNERS
MARRIAGE_BRIDGE
FULL_SIBLINGS
```

```
FATHERS_SIDE_SIBLINGS
MOTHERS_SIDE_SIBLINGS
SIBLING_BRIDGE
```

5.7.2 Sibling Types

The sibling type enum details the type of siblings a bridge represents, these type are:

```
HALF_SIBLINGS
FULL_SIBLINGS
```

The sibling types are used within the calculation of the certainty estimates for the sibling bridge queries. They are used to identify if a sibling bridge is representing a full or half sibling relationship and will then use the correct formula to estimate certainty based upon this.

5.8 Result Objects

During implementation it became apparent that due to the complexity of the query outputs that a class instance would be necessary to store a query's result. Also the careful structuring of the ResultObject class would also lend itself to being able to be passed to other methods to perform analysis or operations on the query output, for example, textual justifications.

An array of result objects is returned when more than one result is found, meaning that a single result object handles only one result of the query.

A results object contains the following information:

Field	**Type**	**Description**
rootLink	Link	The individual given as the parameter to the query.
branchLink	Link	The individual identified as possible solution.
intermediaryLinks1	Link[]	Where more than two edges have been traversed in the graph then these Links are stored in this array.
intermediaryLinks2	Link[]	If a secondary path exists linking the root and branch individual that pertains to the query then the Links of the given path are stored in this array.
supportingSiblingBridges	SiblingBridge[]	Any sibling bridges that support the given result are placed in this array.
supportingMarriageBridges	MarriageBridge[]	Any marriage bridges that support the given result are placed in this array.
failedTestPersonRoot	LinkedPerson	If null then the query has executed, if a person is found here then the query has failed and the individual given in the parameter is found here.
combinedEstimate	float	The combined estimate representing the likelihood of the given query result.
queryType	QueryType	Used for the explanation and labelling of the result object.

The result object offers a full range of expressions and due to its clear encapsulated structure will be able to be passed as an object to other processes to run further analytics upon it.

5.9 Population Queries

In this section an overview of the parameters, output and expected behaviour of the population queries is given. To make use of the population queries it is necessary to create an instance of the class, passing the population to be queried into the constructor. The created PopulationQuery instance can then be called to make queries. This can be done using the below code snippet:

```
LinkedPopulation pop = UseCases.generateNuclearFamilyUseCase13();
PopulationQueries pq = new PopulationQueries(pop);
Utils.printResultSet(pq.getPotentialFatherSideSiblingsOf(3));
```

5.9.1 Get Parent Queries

Method names: **getFatherOf**, **getMotherOf**

Successful Query Output (parameter: p)

Field	**Value**
rootLink	P (given parameter p's corresponding person)
branchLink	possible father/mother
intermediaryLinks1	empty array
intermediaryLinks2	empty array
supportingSiblingBridges	empty array
supportingMarriageBridges	empty array
queryType	FATHERS/MOTHERS

The query finds the person in the population with the given ID p. The set of parent partnerships is then considered in turn with each father/mother found placed into a results object and returned as an array containing all the possible results of the query.

5.9.2 Get Children Query

Method name: **getChildrenOf**

Successful Query Output (parameter: p)

Field	**Value**
rootLink	P (given parameter p's corresponding person)
branchLink	possible child
intermediaryLinks1	empty array
intermediaryLinks2	empty array
supportingSiblingBridges	empty array
supportingMarriageBridges	empty array
queryType	CHILDREN

The query finds the person in the population with the given ID p. The set of childbearing partnerships attached to the person are then considered. Each child within these is placed into a results object and all of these are then returned in an array.

5.9.3 Get Childbearing Partner Query

Method name: **getChildbearingPartnerOf**

Successful Query Output (parameter: p)

Field	Value
rootLink	P (given parameter p's corresponding person)
branchLink	possible childbearing partner
intermediaryLinks1	empty array
intermediaryLinks2	empty array
supportingSiblingBridges	empty array
supportingMarriageBridges	possible marriage bridges joining the root and branch individuals
queryType	CB_PARTNERS

The query finds the person in the population with the given ID p. The set of childbearing partnerships attached to the person are then considered. Each individual on the opposite side of the partnership is placed into a result object. If any marriage bridges exist between the root and branch individual then they are placed into the supporting Marriage bridges array, this also causes for a weighting factor to be added to the combined estimate as defined in the combination function.

5.9.4 Bridge Queries

Method names: **getPotentialMarriageByBridges**, **getPotentialSiblingsByBridges**

Successful Query Output (parameter: p)

Field	Value
rootLink	P (given parameter p's corresponding person)
branchLink	possible spouse/sibling
intermediaryLinks1	empty array
intermediaryLinks2	empty array
supportingSiblingBridges	empty array/currently considered sibling bridge
supportingMarriageBridges	currently considered marriage bridge/empty array
queryType	MARRIAGE_BRIDGE/SIBLING_BRIDGE

The query considers each of the specified bridges of the given type for person P where the persons ID is equal to p. Each person on the opposite side of the considered bridge is placed into a results object with the bridge used in identifying the result placed into the correct bridge array.

5.9.5 Get Sibling Queries

Method names: **getPotentialMotherSideSiblingsOf**, **getPotentialFatherSideSiblingsOf**

Successful Query Output (parameter: p)

Field	Value
rootLink	P (given parameter p's corresponding person)
branchLink	possible sibling
intermediaryLinks1	the edges/links between the root and branch individuals
intermediaryLinks2	empty array
supportingSiblingBridges	possible sibling bridges joining the root and branch individuals
supportingMarriageBridges	empty array
queryType	MOTHERS_SIDE_SIBLINGS/FATHERS_SIDE_SIBLINGS

The queries here make use of a common method getPotentialXSideSiblingsOf into which they pass differing parameters for the potential mother/fathers list of links. From here, each individual in the given list has every other childbearing partnership to which it is attached considered and the linked children of each added to a result object. The additional links between the root link and the branch link are stored in the intermediary links 1. Any sibling bridges that connect the root and branch individuals are placed in the supporting sibling bridges array.

Method names: **getPotentialFullSiblings**

Successful Query Output (parameter: p)

Field	**Value**
rootLink	P (given parameter p's corresponding person)
branchLink	possible sibling
intermediaryLinks1	via father intermediary links
intermediaryLinks2	via mother intermediary links
supportingSiblingBridges	possible sibling bridges joining the root and branch individuals
supportingMarriageBridges	empty array
queryType	FULL_SIBLINGS

The query first retrieves the two result object arrays by calling the father side and mother side queries. The union of these is then considered. If a possible sibling appears in both arrays, it is added to a new result object with the intermediary links from each of the original results object being placed into the new results object. Any sibling bridges from the original results are also placed into the new results object but with duplicates removed.

The queries outlined show a consistent and resilient formatting of results and offer a full complement of standard genealogical step queries, which enables any custom traversal of the structure to be made using a combination of the provided queries.

5.10 Textual Justification

The textual justification class takes in an array of results objects as a parameter and based upon the query type of the set of results outputs text accordingly. The array of result objects are all expected to contain the same query type. The textual justification class makes use of a series of if statements to check for null or empty results sets. In such cases, informative error messages are returned. Otherwise, the method builds a string tailored to the given query type that details each result in the array, including the root and branch individuals, combined certainty estimate, query type, the evidence and supporting bridges. Furthermore in the case of the most likely object in each result object array it is textually described as the 'most likely' with the others described as 'possible'. The method returns a string that the user may then manipulate or read.

A couple of examples of the textual justification can be seen below. The first part of the example is the raw data returned from the query which details the possible query results with a certainty estimation and then the object of persons that are on the traversal, made between the person in the query parameter and the possible results. Furthermore if present the number of supporting sibling bridges (SSB) is detailed.

The text following, beginning ‘The query pertains to…’ is the information returned from the textual justification generator when the array of results objects used to produce the list of possibilities is passed to it.

5.10.1 Children Query Textual Justification Example

```
Possible CHILDREN of d
c @H 0.9 by gamma
a @H 0.8 by alpha
b @H 0.7 by beta

The query pertains to the possible children of the person d.
The most likely children with a certainty estimation of 0.9 is person c (ID: 2) with partnership ID 2 as the joining object, supported by the evidence in records 0 and 6. Person a (ID: 0) is also identified as a possible children with a certainty estimation of 0.8 with partnership ID 0 as the joining object, supported by the evidence in records 2 and 6. Person b (ID: 1) is also identified as a possible children with a certainty estimation of 0.7 with partnership ID 1 as the joining object, supported by the evidence in records 1 and 6.
```

5.10.2 Full Sibling Query Textual Justification Example

```
Possible FULL_SIBLINGS of b
c @H 0.056208774 by d & e with 1 SSB
a @H 0.051787723 by d & e with 1 SSB

The query pertains to the possible full siblings of the person b.
The most likely full siblings with a certainty estimation of 0.056208774 is person c (ID: 2) with persons d (ID: 3) & e (ID: 4) as the common parents, supported by the evidence in records 0, 1, 6 and 3. This sibling bridge is supported by the sibling bridge ID 2 supported by records 0, 1, 6 and 3.
Person a (ID: 0) is also identified as a possible full siblings with a certainty estimation of 0.051787723 with persons d (ID: 3) & e (ID: 4) as the common parents, supported by the evidence in records 2, 1, 6 and 3. This sibling bridge is supported by the sibling bridge ID 0 supported by records 2, 1, 6 and 3.
```

5.11 Use Cases

A wide range of 18 use case examples can be found in the class UseCases. They can be generated by calling the static method for the required Use Case. A LinkedPopulation instance is returned containing the requested use case.

The use cases created are based around 7 initial use cases with the further 11 use case created by adding slight permutations to the some of the initial use cases. Further discussion of the use cases and their expressiveness and usability will be undertaken in the evaluation section.

5.11.1 Creating new use cases

The approach to manually creating use case in code can be seen in the UseCases class. The general outline to doing this follows the steps:

- Initialise a LinkedPopulation instance
- Add LinkedPersons to the population
- Create Evidence
- Initialise ChildBearingParnerships and add to population
- Attach persons to their parental partnership (with Evidence)
- Attach parents to their possible partnerships (with Evidence)
- Initialise SiblingBridges and add to population

- Attach persons to their possible sibling bridges (with Evidence)
- Initialise MarriageBridges and add to population
- Attach persons to their possible marriage bridges (with Evidence)

Test methods to enforce correct construction are detailed in the Tests section.

It is quickly apparent that the construction of these populations becomes time consuming to perform by hand, therefore if we are to build larger testing structures, then a better approach will be necessary.

Already within the Digitising Scotland code base are models for generating large scale populations that are able to produce populations that are representative of specified statistical distributions. However, given the changes made to the interfaces and the need for the LinkedPopulation model to contain evidence and bridge objects the OrganicPopulation generator cannot be used without considerable modification. Also the multiple linkages that exist in the Linked structure will require the population generation model to induce uncertainty and multiple possible linkages into the population. This means in some way simulating the linkage process to some degree, which in itself presents a further challenge which will have to remain beyond the remit of this work for the sake of time. Moreover such a model would want to look at ways of introducing typographical and record error as well as record loss. Beyond simulating the linkage the other option would be to perform the real linkage but as of yet no such algorithms exist to do this to the specification idealised upon throughout. This will be discussed more in the future work section.

5.12 Utils

The Utils class offers a range of utility methods that print various objects such persons, links and result object details to the console as well as offering commonly used functionality throughout the project such as specialised array ordering and joining. The code in of itself should be self-documenting here.

5.13 Tests

The tests provided check to enforce that the populations are consistently structured. This is done by considering every individual in the population and then traversing away from them over every link attached to them to reach all the immediately attached persons. Then for each person all the persons immediately next to them are found. If the initial person is found to be in this set of people then we have demonstrated that the graph is structured correctly between the initial person and all people immediately reachable from them. If we can confirm this for all persons in the graph then we can guarantee all the links and intermediary objects are correctly formed. This is checked for parents' relationships, childbearing partners and for both bridge type.

Testing of querying methods has been performed by hand and comparing the returned queries against the structures as seen with the use cases to ensure the correct results are being given.

5.14 Interfaces

The new interfaces have been outlined and discussed in section 4.9. It was also considered that the Linked population model could implement both the initial and the linked interfaces. However, given that information would have to be lost when meeting the calls in the original interfaces this would lead to ambiguity in the data. For example, returning the list of possible partnership objects without the associated certainty estimates would leave no way of distinguishing between two highly probably partnerships which likely indicate the individual being the member of both partnerships and a highly likely and a much less likely partnership which indicates the person was only likely a member of one partnership in life. This introduction of ambiguity which could lead to the returned data potentially being highly misleading lead to the decision that for the time being that the original interfaces

shouldn't be supported until a way of distinguishing between the two within the original interfaces can be defined. This however is not to say that either set of interfaces are bad, but that they are working with data with distinctly different characteristics. The approach taken to redefine the relationship interfaces however offers a real increase in the expression of the model and it is likely that the Organic model would benefit from also adopting the approach.

5.15 SQL Database Adapter

To ensure populations were being properly created in the design stages before any amount of queries had been written a way to quickly view the created structure was useful. Therefore I adapted the existing MySQL binding code within the project to output LinkedPopulations to a local database. The code to do this can be seen in the adapted_db package within the population representation package and the credit for the original code is to Dr Graham Kirby and Prof Alan Dearle.

To export a linked population to database the following code snippet can be used:

```
LinkedPopulation pop = generateNuclearFamilyUseCase1();
try {
      new ExportPopulationToDB(pop);
} catch (Exception e) {
      e.printStackTrace();
}
```

6 Evaluation and Use Cases

The evaluation under taken here will look at the created model and consider a range of use case and whether it is able to sufficiently express them inline the aims that we initially set out with. It will also consider the scalability of the model is larger scale populations

Obviously a consideration will need to be of the scalability of the model and a short discussion of these will be made and a few points outlined for ways in which perceived issues in this area may be addressed.

6.1 generateNuclearFamilyUseCase

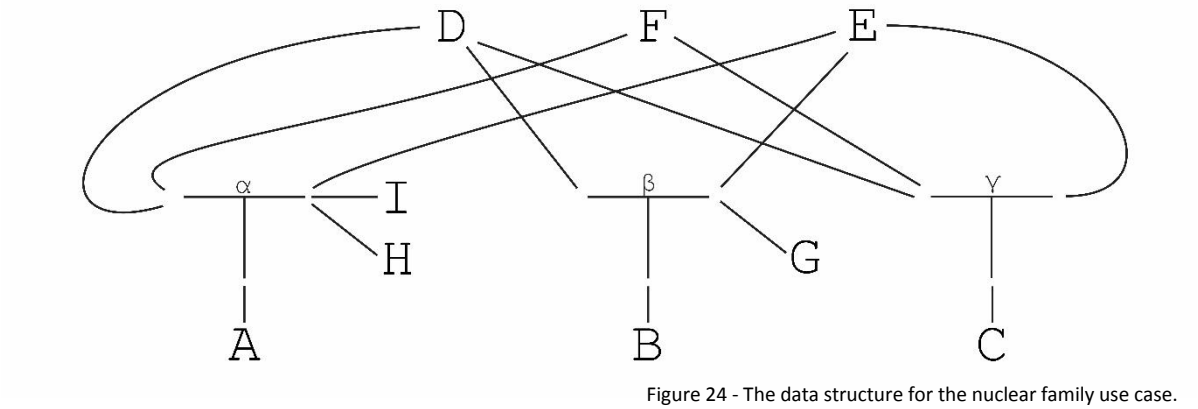


Figure 24 - The data structure for the nuclear family use case.

The nuclear family use case, as shown in figure 24, contains three children, two possible fathers and 4 possible mothers. The uncertainty in the structure means that any combination of these that fits with the given edges could be correct. For example D and E could be the parents of A, B and C giving a standard nuclear 3 child family. However, another possible pedigree could be have F and I as the parents of A and then D and E as the parents of B and C, giving a distinctly different pedigree. The evidence on the edges however allows for us to calculate with a degree of certainty the pedigrees that are more likely.

Other permutations:
Use case 7 - Add sibling bridges between {a,b}, {a,c} and {b,c} (onto use case 1)
Use case 13 - Add marriage bridge between {d,e} (onto use case 7)

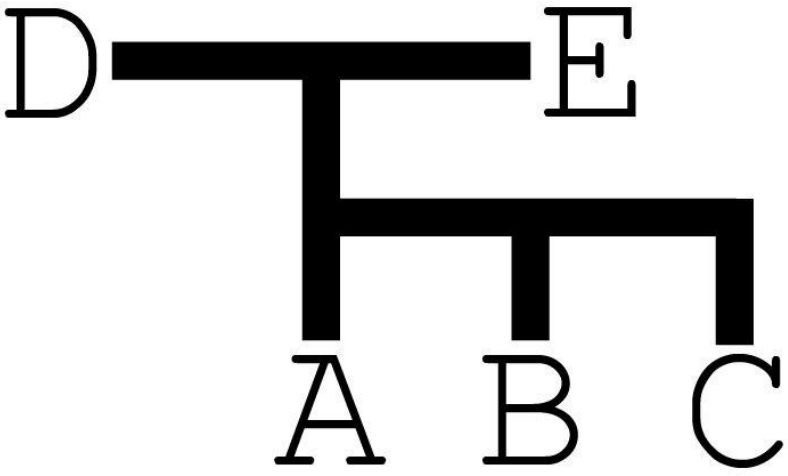


Figure 25 – A family tree representing a possible pedigree of the data structure seen in figure 24.

The bridges in use cases 7 and 13 lead to an expected family structure that potentially has D and E as the married parents of children A, B and C as shown in figure 25.

Use case 8 - Add sibling bridge between {a,c} (onto use case 1)
Use case 14 - Add marriage bridges between {f,e} and {d,g} (onto use case 8)

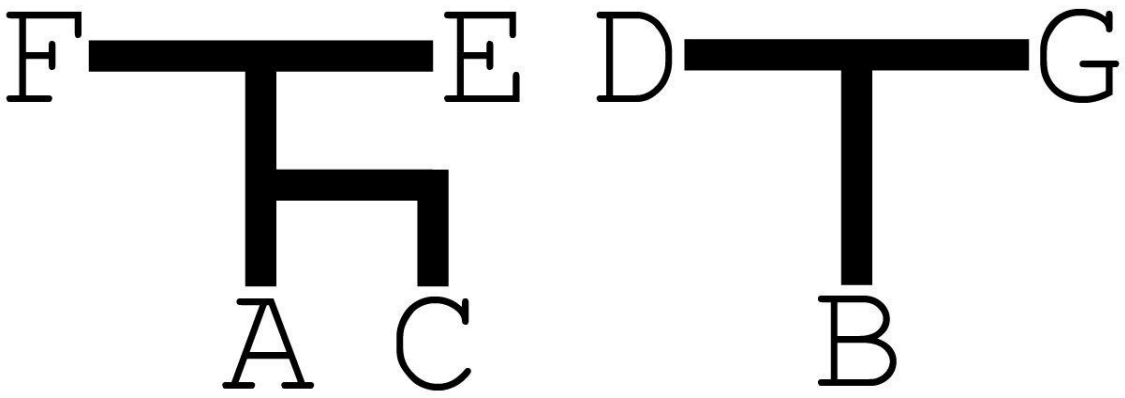


Figure 26 – A family tree representing a possible pedigree of the data structure seen in figure 24.

The bridges in use case 8 and 14 lead to an expected family structure that potentially has F and E as the parents of A and C, and then D and G as the parents of B as shown in figure 26.

6.2 generateNonCrossOverMultiGenerationUseCase2

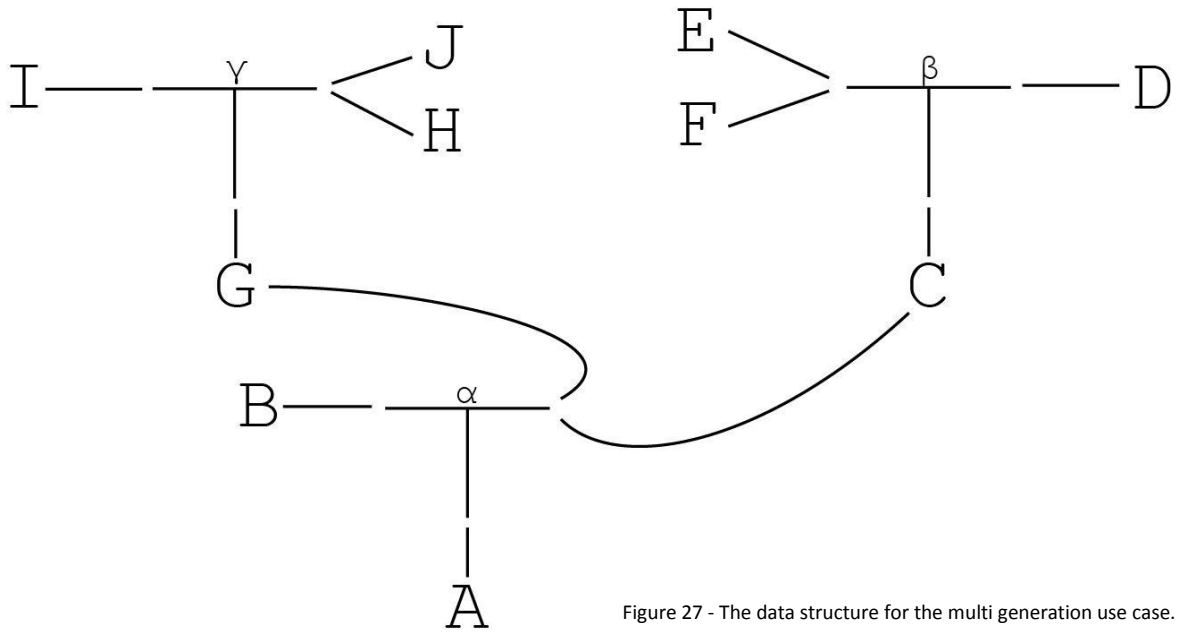


Figure 27 - The data structure for the multi generation use case.

The multi generation use case, as shown in figure 27, contains three generations and in this, the non-cross over case illustrates the possibility that two distinctly different ancestral lines for A can be stored within the structure. To summarise the use case it can be seen that B is the likely father of A with either G or C as the mother, the decision on which is the mother then has onwards implications for the grandparents of A.

Other permutations:

Use case 9 - Add sibling bridge between {g,c} (onto use case 2)

The bridge added in use case 9 gives an idea that G and C may be siblings, however the lack of any cross over between the parents of G and C points to the fact that such a sibling bridge may not pertain to both of them.

Use case 15 - Add marriage bridge between {b,g} (onto use case 2)

The bridge added in the use case 15 gives the idea that B and G are married, here the combined certainty estimate approach of seeing social constructions as indicators of genealogy may indicate the G is now the most likely mother of A.

6.3 generateCrossOverMultiGenerationUseCase3

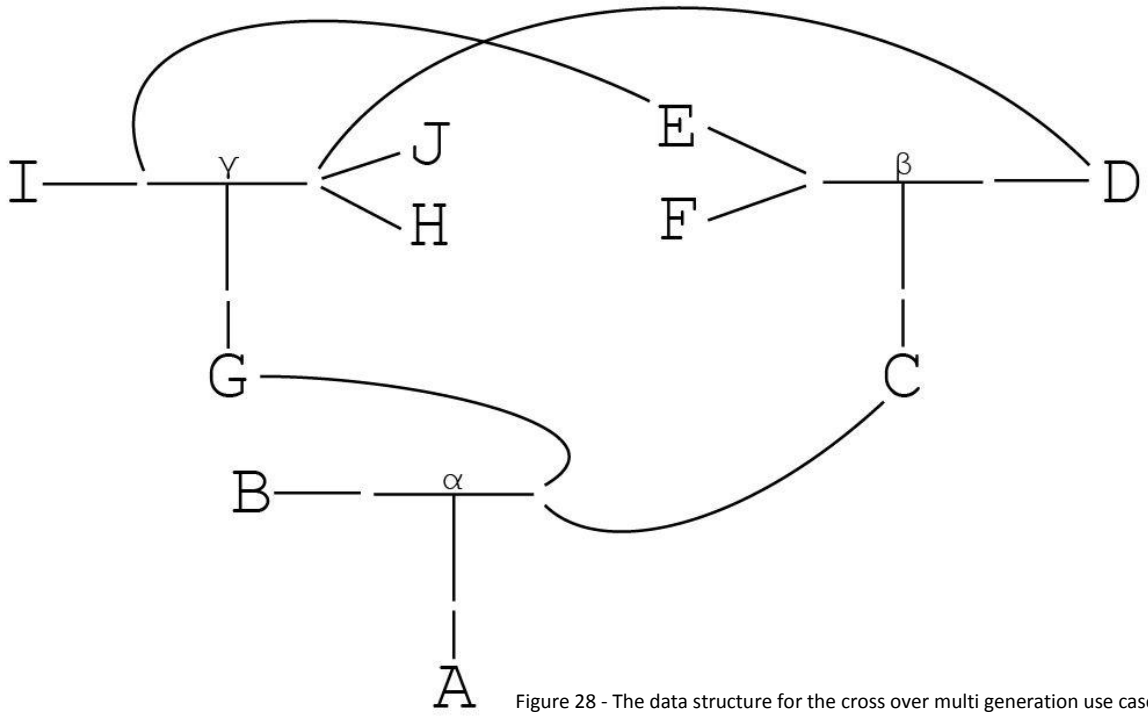


Figure 28 - The data structure for the cross over multi generation use case.

This case, as shown in figure 28, is an extension of case 2 but adds in a set of links which makes it possible that G and C share the same parents - thus potentially making them siblings.

Other permutations:
Use case 11 - Add sibling bridge between {g,c} (onto use case 3)

The bridge added in use case 11 further enforces the idea given in the additional links. Although this may not help for a better decision to be make regarding the mother of A, it does however allow us to be more certain of the grandparents of A (as they are both the parents of G and C) and by extension the construction of a hereditary line through the mother.

Use case 17 - Add marriage bridges between {e,d} and {b,g} (onto use case 11)

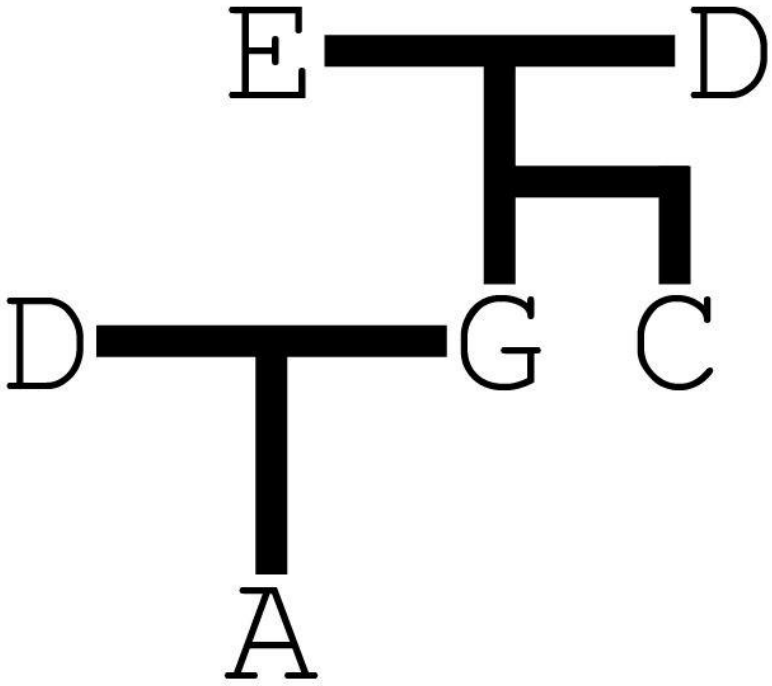


Figure 29 – A family tree representing a possible pedigree of the data structure seen in figure 28.

The bridges added in use case 17 add further support to the idea the E and D are the parents of G and C support by the marriage bridge between them. The second bridge makes suggestion that B and G are married making it more likely that they are the parents of A, as shown in a possible pedigree for this use case in figure 29. However, as a note of caution in using social constructs to further support a supposed linkage it would be interesting to consider such social presumption could be made universally. It is likely that it cannot.

6.4 generateSingleBestFitUseCase4

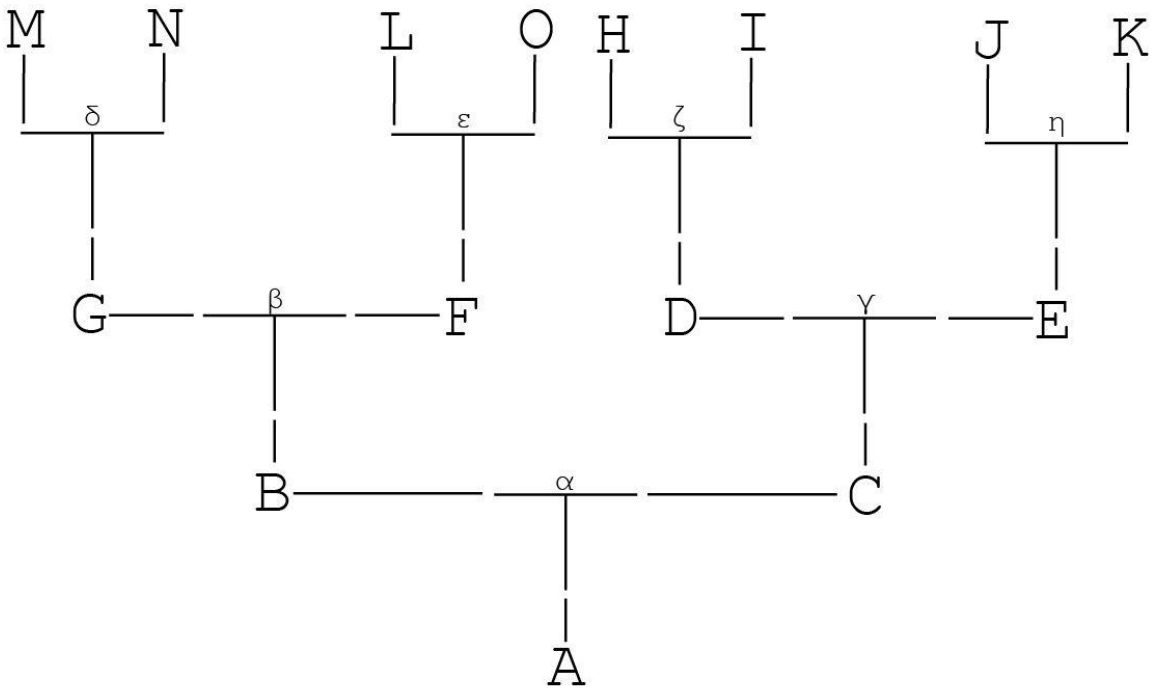


Figure 29 - The data structure for the single best fit use case.

The single best fit use case, as seen in figure 30, offers a standard four generation family structure where single links are offered between each object and person. This case can be seen as a best fit appearance linkage structure and although thoroughly uninteresting due to its lack of uncertainty is important to be able to show the model's ability to express traditional linkage sets within its wider functionality.

6.5 generateMaleLineUseCase5

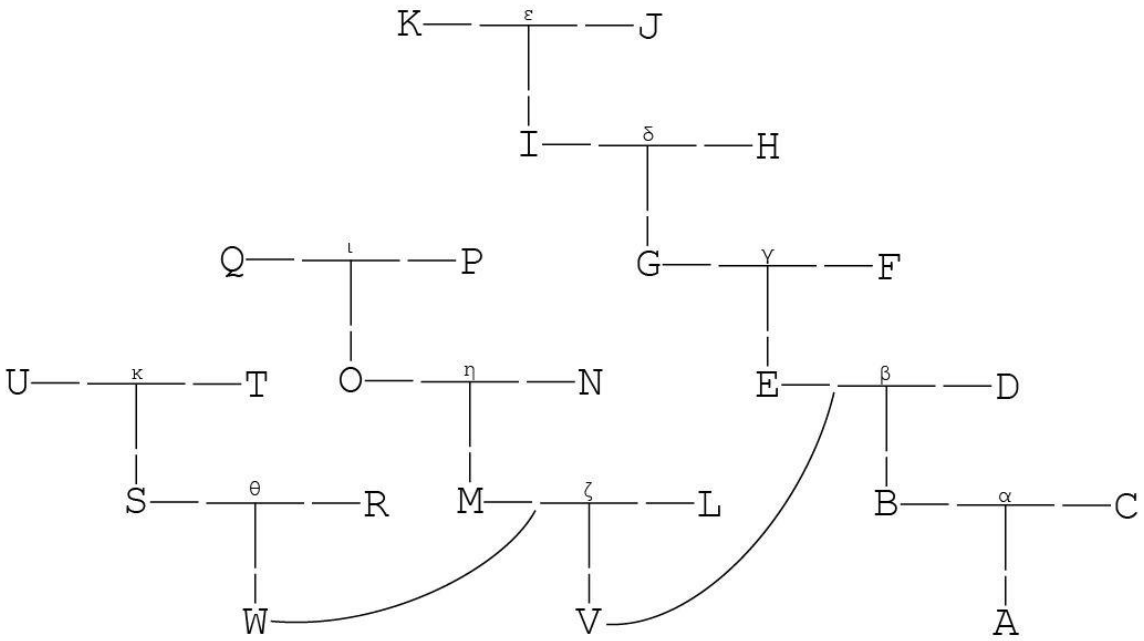


Figure 31 - The data structure for the male line use case.

The male line use case, as shown in figure 31, offers a set of possible 6 generation male ancestral lines for person A. This case study would be useful for testing an interactive approach to querying the structure and the scalability factors that arrive from this.

6.6 generateCousinsUseCase6

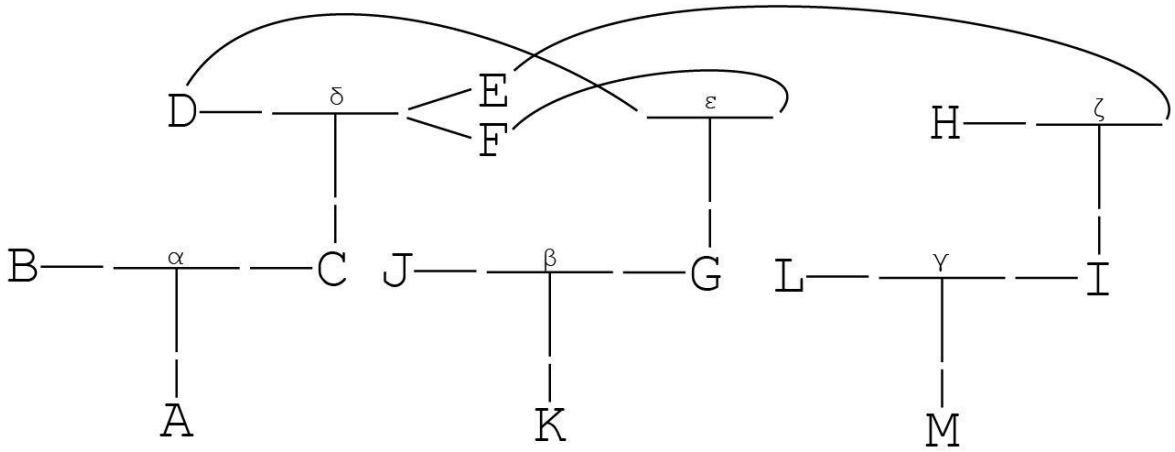


Figure 32 - The data structure for the cousins use case.

The cousins use case, as shown in figure 32, contains three generations allowing for the parents of a person to be considered and then for a traversal across the parent's siblings to be made and then the children of these aunts and uncles to be consider in relation to the initial child. The potential shared

parentage of C and G allows for them to be considered at siblings and therefore their children as cousins to one another.

Other permutations:
Use case 10 - Add sibling bridge between {c,g} (onto use case 6)
Use case 18 - Add marriage bridge between {d,f}, {h,e}, {b,c}, {j,g} and {l,i} (onto use case 10)

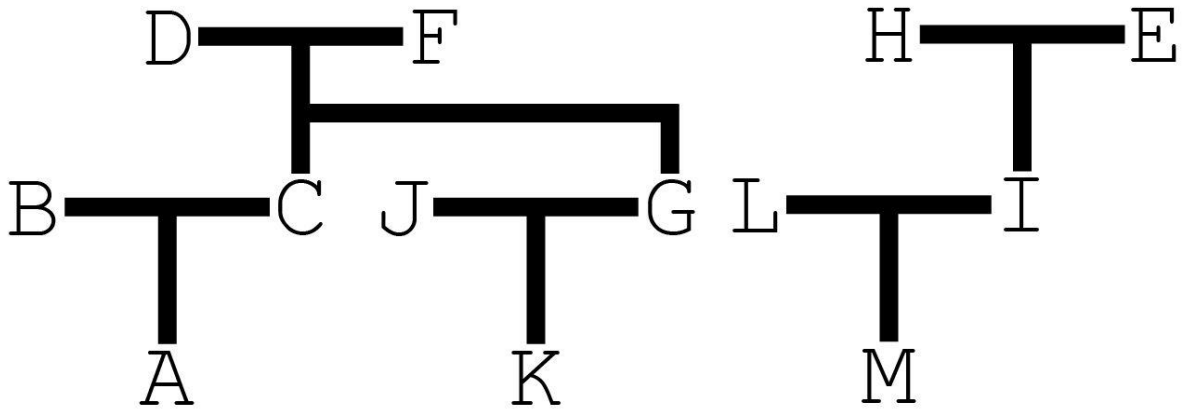


Figure 33 – A family tree representing a possible pedigree of the data structure seen in figure 31.

The bridges added in use case 10 and 18 push towards a possible pedigree consisting of two separate families as is demonstrated in the possible pedigree shown in figure 33.

The above use cases and explanations lay out for us a wide range of ways in which the Linked population structure is able to express a full range of genealogical possibilities. The model in its structuring maintains record dependence by focusing on higher level abstractions of population, for example offspring, marriage and siblings. The model is able to support any number of possibilities attached to the either end of a linking object from none in the case of no data being present to one in the case of a traditional best fit approach to any number of links as is demonstrated through the multiple linkages in some of the use cases. The ability for the model to also support a traditional linkage output as well offers interesting potentials to make use of both traditionally linked data sets and linked data set with uncertainty alongside another in a single query structure. It is also worth noting that the linking objects the structure offers, supports a majority of genealogically significant relationships. Also the options that the marriage bridge approach offers is important as it is an example of how a wider range of records that can be used to imply genealogical relations from the expectations of social constructions. The presence of bridges shows how indirect genealogical records can be used to further inform uncertainties in a structure that is focused on direct genealogical structures.

6.7 Scalability

The scalability of the devised model also needs to be considered. It can be seen from the examples of the structure throughout that the number of permutations that begin to appear in localised structures can increase at a relatively high rate. The volume of possible links and associated evidence could begin to have implications for the memory size of a model, however it is unlikely to cause more than a 2-3x increase in footprint compared to say the Organic Population model. To attain these scaling values it may also be necessary to look into the ways in which evidence records are stored to prevent duplication and the storage of data on parts of records that are unused. The need to be this conservative about memory however is unlikely as the previous population models are able to scale to millions of individuals in reasonable small data footprints (~2GB per million people) and even a threefold increase on this is still manageable.

The complexity of the implemented queries themselves over short spans can be seen to perform well. Issues arising over longer spans will arise in the presence of a high density of possible links as long distance traversals are made. For example, bring me the 15th male ancestor of person P. If we consider that each step offers 3 potential linkages then the number of individuals returned as the possible ancestor is 3^{15} = 14,348,907. For each of these, a certainty estimate will need to be calculated. There are two counters to this issue (although we cannot be sure it is an issue due to not having synthetic data on the scale to be able to test this - the multi-generational queries we can run show no slow down although the longest use case only offers 6 generations) :

1) It is unlikely that we will be making genealogical queries that are this far reaching and are more likely to be focused on more immediate relatives as there will be able to inform us more about the considered individual.
2) By adding the functionality to attach threshold values to queries we can discard some ancestral branches at the point where they cease to become significant or that a large number of better possibilities lie in front of them.

To summarise, the scalability offered by the model is able to express large enough populations in memory to be able to work with population scale data sets. From a complexity viewpoint it would appear that the query methods will be sufficient for making useful queries but that optimisation in areas (thresholds, hash lookups, etc.) may prove a necessary further step to take what is a proof of concept at a small scale to a population scale data set.

7 Conclusion

In conclusion, we will look to outline the discoveries made and the value of these. In addition given this works close reliance and relation to an idealised linkage process, we will also discuss the implications for the design of such a linkage process before moving on to talk about the potential wider applications of the research to other domains. Finally, we will talk about further work, both within and also related to this dissertation.

7.1 Discoveries

The main focus of this work has been to outline a way of structuring genealogical data to allow for uncertainty to be exposed to the end user while still being able to express a full range of genealogical possibilities. This has required us to formulate an understanding of how to incorporate multiple possibly edges of which only one may represent the truth into a structure that maintains integrity and is easily intelligible. The structures shown throughout can be seen to display this level of intelligibility and are able to offer multiple linkages between many individuals that exposes the uncertainty of the underlying linkage to the end user.

The arrival upon the idea of social construct records (i.e. marriage records, social makeup) as indicator for genealogical relationships has been an interesting new viewpoint on viewing different types of records. The use of marriage records as supporting entities in linkage is already widely used but the identification of the idea of using social constructions as a general indicators of genealogical relationships has seen lesser usage. This could be extended to a wider set of records that are also represented of socially motivated indicators of genealogical behaviour (i.e. religion, occupation, salary) which could be paired with an understanding of the variation of the effect of these social factors around the globe. This idea has seen some discussion throughout and the implementation of the combined certainty estimate function provides a way to prioritise more probably links. The occurrence of these calculations at the representation level will require further consideration at the point of implementing the idealised linkage algorithm.

The movement of the divide between linkage and visualisation has been touched on throughout this work. The desire to expose linkage inherent uncertainty at the visualisation level has meant that it cannot be expected that the linkage be wholly encapsulated before the representation level. This has resulted in the opportunity to allow for the user making queries over the linked data to be used in the linkage process. In making these assumptions of the linkage process has allowed for uncertainty to be made visible resulting in many more potential paths across the data structure. Therefore, it has been necessary to consider ways to retain the ability to make useful traversals across the structure when it is required that a more certain pedigree be formed. This now has to be performed at the point of query, in the representation layer, by the use of the combined certainty estimate functions.

7.2 Value

The implications of the discoveries made have the potential to be significant. In short, they offer us the ability to better understand out linked data sets enabling users to use data with a better understanding of the uncertainty in their data. Also in this work we are breaking down any illusions that black box linkage algorithms are a perfectly certain art and exposing to the user the realities of messy data. Hopefully not so that they give up on using this data but that they are able to use it to make better and more informed and proportional decisions in light of the realities of the data. The full realisation of the value of these discoveries will take years of further research before we will see if they truly come to fruition. However, in the meantime the simple presence of research which is looking to expose the messiness of our data may have the effect of improving the awareness of the issues with the data that we work with. Hopefully leading us nearer to a consensus on how large scale data in this day should be produced to minimise these issues.

7.3 Implications for Linkage Process Design

Throughout, we have talked about an idealised linkage process and approaches that a new generation of linkage processes will need to take to allow the exposing of uncertainty at the representation level. The comments made here are in no way complete and simply detail the impressions that we have drawn from considering the problem while producing this dissertation.

- The process will need to identify its reasoning behind each individual linkage.
- The algorithms it uses to indicate certainty will either need to be able to be rerun post linkage or this information will need to be output with the linkage solutions. This may end up having to balance a trade-off between the additional memory footprints of and limiting how wide the linkage algorithms considerations are. Possibly these considerations will be making decisions based on a wide selection of the data that is unfeasible to be output to allow for the rerunning of the uncertainty calculation and so the summarised value alone will have to be output.
- Consideration will have to be given to where the split between the linkages processes ends and the representation certainty estimate calculations begin. It may result in a wholly different approach needing to be found that maximises the amount of work done in the linkage stage that is still able expose some degree of uncertainty to the user.
- A mechanism for dealing with self-linkage due to uncertainty when a linkage occurs between two highly similar individuals
- An enforcement of one-to-one relationships between objects at the source record level and the representation level.

7.4 Application to Wider Domains

The application of the themes of this work to wider domains could have interesting implications across many domains in linkage beyond genealogy. For example, the ability to appreciate and be able to represent uncertainty in data would have real value to the both the health and security domains. Any time we are making decisions based on linked data that effects people we want to be doing so with the fullest understanding and appreciation of our data possible; and in the world of messy data we live in understanding uncertainty in our data is key to that.

7.5 Further Work

At the outset we envisaged that this work would form a reasonably contained project that would be able to exist to a reasonable extent in a vacuum. However, the amount of further work that this dissertation points to being useful is considerable. The need for research in data linkage algorithms which are able to handle uncertainty in large scale linked data sets is most likely the largest. Other interesting areas still requiring further work involve how we move the social constructed proxies into our linkage and uncertainty models effectively, without introducing that many factors of uncertainty, into our models that they become useless. In the way this work expanded out to a complex structure with the expression we required before applying restrictions to that model (i.e. the bridges) to enable us retain an understanding of our data and to better annotate our model, the same will need to be done for each facet of uncertainty we want to involve in our models.

Additionally, within the immediacy of this work, given further time, I would look to implement the query language to the specification discussed, create a model to produce large scale synthetic linked data sets, identify and explore further proxies between social and genealogical events, implement more queries especially focused on many generational queries, query complexity and at a later point to reconsider the availably genealogical ontological schemes and the way in which this work could interact with these.

7.6 Reflection

The process of undertaking a considerable sized piece of research has been a highly enjoyable undertaking. To say it had not been a learning experience could not be further from the truth. The times in my week I have enjoyed the most over the past two semesters have been the times where I have been able to sit down with a white board and my research, and to spend time finding out new things. The project has taught me the value of knowing what you are looking for before you start out and that most of the time the number of lines of code and the number JUnits tests on your CI server are not really all that important in research. These things take on the form of tools, things that have enabled me to do research and prevent the code base from disintegrating when I am more interested in getting code on the page to see if a new idea plays out in reality rather than being bothered about the importance of good code. Furthermore, given how the size changing direction can seem daunting. Making decisions to restructure the code base at one point and then, a week later, revert back to a the more or less the original structuring has shown me that sometimes the only way we can find out if an idea does work is to try it out; but then after that to realise that even though it appears no progress has been made that progress has been made and so not be lose heart in those moments. As Thomas Edison allegedly once said, "I have not failed. I've just found 10,000 ways that won't work" - and I have found in research such a mentality holds true even when it may appear little progress is being made.

7.7 Acknowledgements

I would like to thank my supervisor Dr Graham Kirby for his guidance, help and expertise over the course of this project. I would also like to extend additional thanks to Mr Ben Marshall for the discussions and debates that we have shared which have brought about new ideas about the problems faced; as well as for his time and comments in proof reading this dissertation. I also thank both Mr Ross Creelman and Miss Catherine O'Malley for their time, comments and grasp of grammar far beyond my own, in the proof reading of this dissertation and for their opinions and views on the comprehension of this work to a non-technical reader.

_____

8 References

Aggarwal, C. C. (2009). *Trio A System for Data Uncertainty and Lineage*. In Managing and Mining Uncertain Data (pp. 1-35). Springer US.

Barbará, D., Garcia-Molina, H., & Porter, D. (1992). *The management of probabilistic data*. Knowledge and Data Engineering, IEEE Transactions on, 4(5), 487-502.

Dunn, H. L. (1946). *Record linkage*. American Journal of Public Health and the Nations Health, 36(12), 1412-1416.

Fellegi, I. P., & Sunter, A. B. (1969). *A theory for record linkage*. Journal of the American Statistical Association, 64(328), 1183-1210.

GENTECH. (2000). *A Comprehensive Data Model for Genealogical Research and Analysis* (version 1.1).

Holman, C. D. A. J., Bass, J. A., Rosman, D. L., Smith, M. B., Semmens, J. B., Glasson, E. J., & Stanley, F. J. (2008). *A decade of data linkage in Western Australia: strategic design, applications and benefits of the WA data linkage system*. Australian Health Review, 32(4), 766-777.

McGuinness, D. L., & Van Harmelen, F. (2004). *OWL web ontology language overview*. W3C recommendation, 10(10), 2004.

Melton, L. J. (1996, March). *History of the Rochester epidemiology project*. In Mayo Clinic Proceedings (Vol. 71, No. 3, pp. 266-274). Elsevier.

Newcombe, H. B., Kennedy, J. M., Axford, S. J., & James, A. P. (1986). *Automatic linkage of vital records*. In Record linkage techniques, 1985: proceedings of the Workshop on Exact Matching Methodologies, Arlington, Virginia, May 9-10, 1985: co-sponsored with the Washington Statistical Society and the Federal Committee on Statistical Methodology (Vol. 1299, p. 7). Dept. of the Treasury, Internal Revenue Service, Statistics of Income Division.

Stevens, R., & Stevens, M. (2008). *A Family History Knowledge Base Using OWL 2*. In Proceeings of OWL: Experiences and Directions Workshop (OWLED 2008).

Tsarkov, D., Sattler, U., Stevens, R., & Stevens, R. (2009, October). *A Solution for the Man-Man Problem in the Family History Knowledge Base*. In OWLED (Vol. 529).